

DAVID BRAVO

ANIMAL CROSSING

GUITAR HERO III

RARE

Y MUCHO MAS...

Nº 6 - SEPTIEMBRE 2008
PUBLICACION GRATUITA





STAFF:

Alain Ibáñez de Opakua – alain@altmagazine.net



Antonio Garrido – antonio@altmagazine.net

Fernando Vergel – nandovergel@gmail.com



Pablo Vilches – pablo@altmagazine.net



Ester Sánchez – ester@altmagazine.net

Saioa Urresti – saioa@altmagazine.net



Jon Agirre – jon@altmagazine.net



Israel López – israel@altmagazine.net



Aritz Durana – aritz@altmagazine.net

Iker Unai Larburu – iker@altmagazine.net



Veronica dell'Oro – veronica@altmagazine.net



Pablo Cervantes – cervantes@altmagazine.net



Ocularis – http://ocularis.es/

Sandy – http://digitalskies.net/

MAQUETACION





Este es nuestro 6º número, y han cambiado algunas cosas en nuestra revista. Hemos cambiado (ligeramente) algunos formatos, hemos adaptado ciertas cosas que se podían mejorar, etc. Durante nuestra andadura también ha habido bajas y nuevas incorporaciones. En este caso hemos tenido que decir "adiós", o mejor dicho, "hasta luego" a uno de los colaboradores que llevaba con nosotros desde el principio, Manuel Sagra. Pero como decimos, es un "hasta luego", ya que esperamos que de vez en cuando nos haga un hueco en su agenda para alguna colaboración esporádica. Y a su vez, damos la bienvenida al Dr. Aritz Durana, mago de la química, francotirador en los *action-RPG* y capaz de pasar *a través del fuego y las llamas*, al que hemos engañado con la excusa de una simple colaboración mientras que nuestra verdadera intención era que se quedase con nosotros por mucho tiempo (*I tricked you! I tricked you!*, que diría *Sue*, del *Grandia*).

Y sin duda la gran novedad es que, viendo la enorme aceptación que tiene esta publicación, y puesto que por motivos logísticos no llega a tanta gente como quisiéramos, hemos decidido abrir un *blog*. En él publicaremos noticias que sean poco intemporales, o que no tengan la longitud apropiada para la revista en sí. Además, este *blog* intentará dar un poco más de ritmo al contacto entre nosotros y nuestro público, principalmente por la facilidad para poder dejar mensajes. Os avisaremos oportunamente a través de nuestra web de la apertura de este espacio.

Y pasando ya al contenido de este número, esta vez vamos a someter a debate si los juegos *"Training"* tienen una utilidad médica real. ¿Vosotros qué pensáis? Hemos analizado este tema, y los usuarios del foro de *gp32spain* vuelven a colaborar con algunas opiniones y una completa estadística. Además, gracias al gran trabajo de Alain, hemos podido contar con la colaboración estelar de *Ocularis*, oftalmólogo en la vida real y personaje insigne de la *blog-o-sfera* científica, sobre cuya aséptica visión hemos podido formar nuestra opinión.

Hablaremos de las genialidades y excentricidades de Richard Garriot (alias *"Lord British"*) en una lectura muy recomendable, y por otro lado relataremos la historia de *Rare* desde su nacimiento.

La *GP2X* es una gran máquina, creemos que ya lo hemos dejado claro. Por ese motivo muchos de vosotros podéis estar pensando en iniciar vuestro propio proyecto, pero no sabéis como empezar a pesar de vuestros conocimientos de programación. Bueno, pues trataremos de echaros una mano con un tutorial.

No olvidéis leer la entrevista a David Bravo, autor de *Copia Este Libro* (de cuyo contenido os hablamos en el anterior número). Sus conocimientos de los derechos del usuario en la sociedad de la información digital (no obstante es abogado) y su lucha dialéctica contra el canon han motivado que nos pusiésemos en contacto con él y, a pesar de sus infinitos compromisos, ha contestado a todas y cada una de nuestras preguntas. ¡No dejéis de echarle un vistazo!

Dentro del capítulo de colaboraciones especiales, tenemos el gusto de presentar al experto modificador de consolas Fernando Vergel, que nos relata las diferencias entre los modelos de *Master System*, piedra angular en el puente vital de *SEGA*. Y para nuestra sección de arte hemos escogido una obra de *Sandy*, una incipiente artista australiana muy aficionada a los videojuegos y al anime.

Y si sois seguidores de la franquicia *Animal Crossing* de *Nintendo*, es completamente imprescindible devorar el extenso artículo que disecciona este original juego.

Hay muchas más cosas que os gustarán, como *Guitar Hero III* o *Fluxus*.

Pensamos que el *set list* de este número vuelve a ser espectacular, así que tan sólo esperamos que lo disfrutéis y os sea de utilidad.





JUEGOS "TRAINING": ¿SIRVEN PARA ALGO?

La salud es algo que la gente se toma muy en serio. Se trata de un artículo de primera necesidad, por lo que el usuario raramente va a escatimar gastos si cree que un producto le va a ayudar a que esté más sano. Pero a diferencia de la ropa, la comida o la vivienda, normalmente el potencial cliente tiene conocimientos limitados de medicina para conocer la necesidad real que tiene del producto que le intentan vender. Como la salud vende y el cliente está bastante desprotegido, hay leyes que nos amparan, impidiendo un mercantilismo de los productos sanitarios. Por eso vemos tan pocos anuncios de medicamentos, excepto unos pocos productos (para los catarros, etc). Los médicos son los encargados de prescribir los medicamentos, y se restringe la publicidad para fármacos que requieren receta médica.

Pero queda un resquicio legal: los productos que no son sanitarios, que no mejoran nuestra salud, pero que damos a entender que sí. Esa estrategia le ha ido muy bien a *Danone* (no, *Actimel* NO mejora nuestras defensas). Ahora le toca el turno al sector del entretenimiento.

Training for your eyes, entrena y relaja la vista, reza el eslogan de la gran campaña publicitaria que ha lanzado Nintendo para promocionar su nuevo videojuego. Unos pocos ejemplos del texto utilizado para el marketing:

- ¿Con qué frecuencia ejercitamos la vista? Pese a que nuestra vida cotidiana depende de la vista, ¿a que no se nos ocurre practicar ejercicios físicos con los ojos?
- ¿Sabías que la capacidad visual tiene cinco aspectos distintos que se pueden fortalecer? *Training for your eyes* dispone de una amplia variedad de ejercicios diseñados para cada uno de ellos.

Las explicaciones dan una imagen de seriedad “científica”, y también mencionan a un tal Dr. Hisao Ishigaki, que nos lo presentan como experto en el tema. Después de un vistazo, es fácil que lleguemos a la conclusión de que el *Training for your eyes* es algo más que un videojuego para entretenernos. Se trataría de una forma de entrenar nuestros ojos para potenciar y mantener nuestras capacidades. Para aquellas personas con gafas o problemas en la visión, ¿no sería útil mantener

unos ejercicios gracias al *Training for your eyes*?

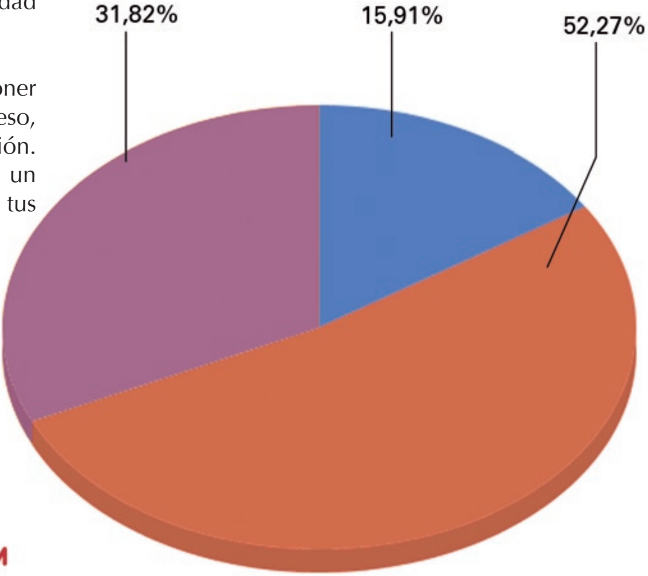
Nada más lejos de la realidad. En el anuncio de televisión la legislación les obligó a poner que este producto no sirve para tratar enfermedades oculares, y ya podemos intuir por dónde van los tiros. La campaña publicitaria está muy bien diseñada, bordeando hábilmente el límite de la publicidad engañosa. Nos dicen que por qué no se nos ocurre practicar ejercicios con nuestros ojos, pero no afirman explícitamente que sea bueno hacerlos. Nos dicen que hay 5 aspectos que se pueden fortalecer, y que *Training for your eyes* tiene varios ejercicios diseñados para ello, pero tampoco afirman que efectivamente se potencien estos aspectos con el videojuego. Nos presentan a un doctor japonés experto en el tema, pero no es oftalmólogo, ni siquiera médico (que sería el que realmente pudiera aconsejarnos sobre temas de la vista, ¿no?).

Y es que, *Training for your eyes* no potencia ninguna habilidad del globo ocular. Nuestros ojos no necesitan entrenamiento, ya los estamos entrenando todo el día. Como mucho, necesitan relajarse del enfoque cercano si trabajamos mucho con ordenador o leyendo. Y estar mirando la pantalla de la *Nintendo DS* no es precisamente la mejor forma de relajar los ojos.

Sí que puede potenciar algunos aspectos concretos del sistema visual, pero no de los ojos. Para memorizar rápidamente números, para "cazar" objetos en la pantalla. Pero esas propiedades no son propias del ojo, sino más bien del cerebro que interpreta las imágenes. *Training for your eyes* tampoco nos va a ayudar a realizar mejor actividades cotidianas como leer o conducir (como nos quieren hacer entender en su publicidad).

¿Entonces *Training for your eyes* es malo para los ojos? Tampoco. Es más, personalmente me parece un juego entretenido. Pero es eso, un juego. A costa de repetir los ejercicios, vamos mejorando, pero sólo en las rutinas concretas del ejercicio en cuestión, de la misma manera que vamos mejorando en habilidad cuando jugamos a un *shooter*.

Por tanto, si te gustan los juegos en los que poner a prueba tus habilidades e ir controlando tu progreso, *Training for your eyes* puede ser una buena opción. Pero siempre teniendo bien claro que es eso, un entretenimiento. No estás comprando "salud para tus ojos".



VOTACION REALIZADA EN WWW.GP32SPAIN.COM

por el Doctor Pablo

Nintendo ha dado en el clavo, comercialmente hablando, con estos juegos. Es un concepto que ya sabíamos que estaba ahí, pero que nadie había conseguido explotar de una forma adecuada. Es decir, nadie se había planteado un marketing serio ya que se ponía en duda el interés de los jugadores por esta clase de productos. Justamente ahí encontramos la clave, en la forma de venderlo.

Quiero decir que Nintendo empezó a olisquear que estos productos, si bien no iban a ser consumidos masivamente por los jugadores tradicionales, sí que podrían llegar a un sector inexplorado hasta ahora y mucho más numeroso: el auténtico y genuino jugador casual. Y la mejor forma de vender todo esto es utilizando la misma publicidad engañosa que voluntariamente nos podemos tragar todos los días. Pero hay una diferencia que pone en ventaja a estos juegos frente a los anuncios de detergentes, puesto que todos sabemos que nadie se traga esas escenas con las señoras en la plaza del pueblo aplaudiendo por cómo desaparecen las manchas de la ropa (con música de feria de fondo). En cambio, con estos productos tan novedosos para ese gran banco de peces existe un alto nivel de desconocimiento. El anzuelo funciona.

Son productos positivos, que ofrecen experiencias enriquecedoras, puede que incluso necesarios para la evolución y manutención de la industria, pero no te van a convertir en alguien más listo y saludable.

por el Doctor Iker

Primero una aclaración. El público potencial de este tipo de productos (léase: Brain Training, Big Brain Academy, el mencionado Training for your eyes…) creo que puede dividirse en dos grandes grupos: 1) aquellos que lo juegan por puro ocio conscientes de que el producto en cuestión no les va a entrenar más que un Mario Party y 2) aquellos que realmente ven el juego como un elemento preventivo de posibles "dolencias".

Casi podemos decir que el primer grupo lo conforman los jugadores tradicionales, que quedamos, en principio, fuera de esta reflexión. Al fin y al cabo, si nos encaprichamos de un juego nos importará entre poco y nada el anuncio correspondiente (si exceptuamos que puede proporcionarnos información como la fecha de salida).

La clave está en el segundo grupo, el que compra "salud" o "prevención". Aquí, ¿que quieres que te diga?, las leyes del mercado imperan. Si tú, como proveedor, eres capaz de provocar en el consumidor una sensación de necesidad, me parece perfecto que la explotes.

Todos somos borregos en algún momento y estamos dispuestos a soltar nuestro dinero por algo cuyo valor es dudoso comparado con el beneficio que nos va a aportar. No hace falta volver al típico vendedor de remedios milagrosos que aparece en algunas películas, que vende un elixir que cura hasta el aburrimiento. Muchos comprarán el desodorante AXE para poder quitarse a las mujeres de encima, crema de baba de caracol para quitarse las manchas y arrugas de la piel, BMW por el prestigio de tener uno, …

A mí me parece que es totalmente legítimo que se use cualquier tipo de asociación de las antes mencionadas para vender un producto, siempre y cuando éste no cause daño al consumidor final. Imaginad un anuncio que rece algo así como: "El 50% de las personas mayores de 45 años sufren de dolor gástrico. Se ha comprobado que el consumo de arsénico elimina el dolor gástrico en pocos minutos…" Seguro que más de uno lo compraría.

por la Doctora Saioa

Quizá no esté totalmente demostrado que estos juegos mejoren las cualidades de la gran mayoría de gente que juega a ellos, que aun ahora, y pese a los esfuerzos de Nintendo por integrar a más tipos de gente, sigue siendo predominantemente (que no exclusivamente) el habitual. Pero al parecer, sí que se ha demostrado que ciertos colectivos, como la gente con enfermedades neurodegenerativas (alzheimer, etc.), o gente mayor, o incluso niños que están aprendiendo a sumar y restar (por poner unos ejemplo), son más receptivos, y estos juegos les ayudan a mejorar.

El problema está en la campaña de marketing que sigue Nintendo para venderlos al público general, que roza la línea de la publicidad engañosa. No me parece muy lógico que a actores como Javier Cámara, Amparo Baró o Nicole Kidman el juego les adjudique tanta edad como aparece en los anuncios, teniendo en cuenta que por su profesión se supone que tienen una elevada retentiva y capacidad de improvisación (que son algunas de las cualidades que puede favorecer el realizar correctamente algunos ejercicios mentales de los Brain Training y demás).

por el Doctor Jon

Yo creo que dentro del grupo de juegos training se pueden hacer dos categorías: los que sirven para algo y los que no. De los que no sirven para nada (divertir quizá) no voy a hablar, pero dentro de los que sirven para algo, podemos encontrar la interesante colaboración entre Nintendo y el Dr. Ryuta Kawashima (autor de varios artículos en el área de gerontología que tuve a bien comprar para evaluar la validez de Brain Training en el momento de su publicación), cuyos frutos (y consecuencias) se siguen recogiendo a día de hoy.

Jugar a Brain Training no va a elevar tu mediocre coeficiente intelectual, ni va a provocar que de repente rechaces ver telebasura, ni va a conseguir que desarrolles la tan ansiada telequinesia que te permitiría jugar y hacer la cena a la vez. Los ejercicios propuestos en Brain Training han mostrado su eficacia en retardar parcialmente el deterioro cerebral en pacientes con enfermedades neurodegenerativas como el alzheimer. Claro, cualquiera puede ejercitar su cerebro de esta forma, pero dado que seguramente no estarás enfermo ni serás especialmente mayor, no te va a servir para mucho más que calcular cuánta pasta te debe un amigo o leer a toda velocidad tus foros preferidos.

En resumen: juega a algo porque te lo pases bien, y cuando te duela algo, ve al médico. Y mejor que entrenar tu cerebro para suplir el neurocidio que le ocasionas con las borracheras o viendo televisión, haz deporte, que eso sí ayuda a calmar la ansiedad y el estrés.

Pakoito

Sirven y no sirven: no vas a mejorar ni te van a convertir en un supergenio, pero ayudan a la gente que no ha resuelto dos sumas en su vida a comerse un poco el coco, lo que por definición les hace ejercitar la mente. Por una vez. Ya podría haber un "educación trainer" o un "formación de criterio trainer".

otto_xd

A mi los english si me sirven, aparte de ver series y tal, me perfeccionan el oido (que lo tengo muy malo), aprendo un monton de vocavulario y me fijo en como se escriben las oraciones (mi gramatica en ingles es peor que en castellano)

Los Brain creo que dan agilidad en el calculo y en reflejos, pero no te hacen mas listo, eso si, incluso podria servir para espabilar ciertos cerebros.

Mariner_7

En el caso de los English Training pues la verdad es que es como todo curso de inglés pero en una consola, no vale más que para tener un poco de vocabulario muy básico y poco más. Se basan en lo mismo que los típicos cursos de "domine el ingles con menos de 1000 palabras": querer saberlo todo con poco esfuerzo.

En el caso del Brain Training, aparte que yo de neurología ni idea, creo que se podría basar en una falsa percepción. Me explico, si todos los días hago cuentas, que es lo que hace, a la hora de calcular una cantidad seré mas ágil. Es como la gimnasia, si corro todos los días cuando haga una maratón pues lo haré mejor que si estuviera tirado en el sofa. Eso es lo que te hace creer que tu cerebro está "en forma", pero a veces al que entrena corriendo todos los días le dan infartos.

En el que no estoy nada de acuerdo es en el Training for your Eyes. Los otros tienen un mínimo interés y pueden dar resultados, pero la forma de vender el TFYE no me la creo. Lo venden diciendo que todo el mundo trabaja con ordenadores y la vista sufre, y no hay nada mejor para arreglarlo que mirar otra pantalla de 3 pulgadas. En vez de gastarte los 30 euros del training mejor es ir a darse un paso, tomar el fresco y admirar el paisaje que es gratis y si entonces tus ojos no descansan es que tienes presbicia.

Jurk

Cada vez que juego a algún juego de estos me acuerdo de los cuadernillos Rubio y de Vacaciones Santillana. Sí, aquellos librillos que solía comprar tu madre para que en verano repasases lo que estudiaste durante el curso. Sólo que esta vez vienen en formato electrónico.

Los training funcionan exactamente como esos cuadernillos: no hay contenido nuevo ni nada que no supieras antes, te lo presentan en un formato atractivo y ameno y su uso diario sirve para afianzar y entrenar sobre lo que ya sabes. Incluso incluyen una madre que te echa la bronca por no hacer los deberes en forma de profesor Kawashima.

Del mismo modo en el cual ningún cuaderno Santillana te explicaba conceptos del siguiente curso, ninguno de estos juegos te va a volver más listo ni tu vista se volverá la de un águila.

En mi humilde opinión creo que son una buena herramienta de mantenimiento. Y muy divertidos... me voy a jugar al Brain Training.

dardo

Todo lo que sea practicar la memoria es bueno, porque la verdad es que se atasca.

Ahora que no me tengo que aprender de carrerilla listas de cosas como cuando era pequeño, creo que me cuesta mucho más ejercitar la memoria.

Antes nos metían tralla con los reyes godos, las tablas de multiplicar y cosas de esas, pero es que los niños de ahora no hacen una suma sin calculadora, y se aprenden los pokemones (que ningún chulo me venga a decir que oficialmente se dice igual en singular o en plural) después de cincuenta horas de juego. Antes aprenderte una lista de 150 elementos para recordarla nos llevaba dos tardes.

Creo que de todas maneras aunque practicar la agudeza visual (o más bien la inteligencia visual), hacer series de números rápidamente es útil (cual es el sucesor y esas cosas) y sumar de cabeza números de seis cifras sin olvidarte de cuanto valían las unidades cuando ya vas por las decenas de millar desde luego no te hace más inteligente.

Quien practica series o números es más hábil con ellas, igual que agrupar formas y colores. Es como la mayoría de juegos de estrategia. Realmente no hay ninguna estrategia, suele ganar el que conozca mejor la mecánica del juego.

pybey

Mi opinión es según de qué training estemos hablando.

Por ejemplo el Big Brain, pues creo que más listo no te hará, pero sí te da más agilidad para cálculos. Es de lógica que si todos los días haces algo, llega un momento en que tu mente está acostumbrada y lo hace mas rápido y mejor con menos esfuerzo.
Ahora, en el de los ojos, la verdad que intentan venderlo con la premisa que te mejora la percepción los reflejos y no sé qué más, pero en una pantalla tan pequeña ¿como mejoras los reflejos si no hace falta mover los ojos?

En el del inglés yo también lo veo muy practico, por lo menos yo, que mi nivel de inglés es muy básico, algo sí he conseguido, y es saber escribirlo algo mejor y para escucharlo también. Aunque no es una mejoría para tirar cohetes, sí hace algo.

De todas formas, yo estos juegos los cojo con muchas ganas y estoy una o dos semanas haciéndolo todos los días y luego me canso y los dejo apartados, porque al tiempo eso de tener que buscar 5 o 10 min por obligación como que no me hace gracia. Yo quiero distraerme con la consola, no que tenga una obligación con ella.

dj syto

Yo lo que opino es que a gente medianamente inteligente no les va a servir para nada.

Supongo que serían unos buenos ejercicios para gente más joven, o gente con inteligencia bastante limitada o poca cultura. Pero el problema es que esas personas no suelen molestarse en conseguir este tipo de juegos.

O sea que, ¿para que sirven? Pues para vender consolas y juegos.

POR ISRAEL LÓPEZ

RICHARD GARRIOT: LORD BRITISH

Hablar de Richard Garriot no es sólo hacerlo de uno de los mayores talentos que ha dado la industria de los videojuegos, sino de un visionario, un aventurero capaz de cruzar los polos e incluso viajar fuera de nuestro planeta.



Richard Garriot, señor de Britania



Richard Garriot nació en 1961 en Cambridge, Inglaterra, aunque se crió en los Estados Unidos. Hijo del famoso astronauta Owen Garriot (quien batió todos los records de estancia fuera de la Tierra pasando 60 días en la estación espacial *Skylab*), se sintió fascinado muy pronto por las estrellas.

Fue en el instituto, donde sus compañeros le pusieron el apodo de *Lord British* por su refinado inglés que les hacía creer erróneamente que era acento británico. En esta misma época empezó a programar sus primeros juegos con tan sólo 14 años.

Cuatro años después, en 1979, creó el que sería su primer juego publicado: *Akalabeth*. Trabajando como dependiente en una tienda informática, programó este juego en el *Basic* del *Apple II* y, animado por el dueño de la tienda, se atrevió a venderlo. Para ello invirtió 200$ en blisters, carátulas y manuales, y llegó a vender 8 unidades, una de las cuales afortunadamente llegó a las manos de una distribuidora (*California Pacific*) que se ofreció inmediatamente para comercializarlo, reportándole bastante dinero.

Con su primer juego recién publicado y de forma exitosa, empezó sus estudios en la Universidad de Texas, lo que no le impidió crear el primer título de una de las más importantes sagas de los videojuegos: *Ultima*.

En 1982 acabó *Ultima II*, abandonando el *Basic* y programándolo ya en lenguaje ensamblador para aprovechar mejor el *Apple II*. Debido a problemas económicos con *California Pacific*, empezó a buscar otra distribuidora, decantándose finalmente por *Sierra On-Line*, que fue la única que accedió a su exigencia de incluir junto al juego un mapa impreso en tela.

Un año después, en 1983, se decidió a invertir el dinero ganado en fundar junto a su padre y su hermano su propia compañía, *Origin Systems*, para encargarse de la publicación y distribución de sus futuros juegos.

Con *Origin* recién fundada publicó *Ultima III*, pero su verdadero salto de calidad llegó en 1985 con *Ultima IV*. En los tres episodios anteriores se seguía el típico argumento del héroe que tiene que matar al malvado supremo, pero para este nuevo capítulo de la saga introdujo connotaciones éticas y morales asociadas a las 8 virtudes, personajes



más realistas que no se movían en arquetipos de buenos y malos sino en lugares intermedios, y un guión mucho más trabajado.

En 1988 aparece una nueva entrega de la saga, *Ultima V*, pero no sería hasta 1990 con *Ultima VI* cuando daría un nuevo vuelco a la serie con la inclusión de una perspectiva isométrica, gracias al salto tecnológico que supuso abandonar el *Apple II* en beneficio de los nuevos PCs.

En 1992 publica la que sería la primera parte de su gran obra maestra: *Ultima VII: The Black Gate*. Ese mismo año supuso la venta de *Origin* a *Electronic Arts*, y un año después saldría a la venta la segunda parte, *Ultima VII: The Serpent Isle*.

A partir de su integración en la estructura de *EA*, las imposiciones de la editora hacen que decaiga la calidad de sus productos, como el criticado *Ultima VIII* (1994). No obstante, Richard, junto a su equipo, daría un nuevo bombazo en 1997 con la publicación de *Ultima Online*, uno de los primeros y más exitosos juegos multijugador masivos que se haya creado nunca.

En 1999 publica *Ultima IX*, pero la calidad del producto final deja mucho que desear por las prisas metidas por *EA*. Este motivo, junto a la cancelación de varios proyectos de Origin (con *Ultima Online 2* como detonante), provoca que Richard abandone *Origin* y funde en el año 2000 su nueva empresa: *Destination Games*.

Un año después, en 2001, Richard anuncia un acuerdo con el gigante surcoreano *NCsoft*, empresa especializada en juegos multijugador masivos online y creadora entre otros de *Lineage*, por el que *Destination Games* actuará como su sucursal en los EEUU, cambiando su nombre a *NCsoft Austin*.

Tras ejercer como productor en varios juegos de *NCsoft* (*Lineage II*, *City of Heroes*...), su vuelta a la popularidad por la puerta grande sería gracias a *Tabula Rasa* (2007), uno de sus proyectos más personales, donde ha invertido años ideando el mundo en que se ambienta, sus habitantes, lenguaje pictórico, etc. Además, vuelve a retomar los conflictos éticos que tan famosos se hicieron en la saga *Ultima*.

Richard Garriot, el aventurero

Si bien dentro del mundo de los videojuegos Richard Garriot es una celebridad por su saga *Ultima*, fuera de él no se queda atrás y se ha hecho un nombre por su espíritu aventurero, emprendedor y filantrópico.

Sus éxitos en los videojuegos le han convertido en un hombre con una fortuna bastante apreciable, suficiente para construir una mansión de más de 20 millones de dólares llamada *Britannia Manor* (cuyo nombre viene del castillo de *Lord British* en la saga *Ultima*)

o el nuevo castillo que está edificando.

Britannia Manor fue diseñada siguiendo la arquitectura mostrada en la saga *Ultima* con su observatorio, pasadizos ocultos e incluso unas mazmorras. No solo eso, sino que se convirtió en una de las "casas encantadas" más famosas de los EEUU, en la que organizaba tours gratuitos en las fiestas de *Halloween*.

Su afición por la magia y la antigüedad le ha llevado a coleccionar montones de objetos únicos como armas antiguas, extraños artefactos y una serie de objetos espaciales comprados a Rusia, que incluyen un traje espacial, un vehículo lunar (que se encuentra en la Luna) ¡e incluso un *Sputnik* original!

Como miembro del "Club de los Exploradores", Richard Garriot ha realizado exploraciones tan dispares como la búsqueda de meteoritos en la Antártida, el seguimiento de gorilas en las montañas de Rwanda o visitas al Titanic y fallas submarinas.

Por otro lado, su pasión por el espacio, inculcada por su padre, le ha hecho implicarse en varios proyectos relacionados.

Es uno de los principales inversores de *Zero G*, una compañía que organiza experiencias de ingravidez para sus clientes. También dona una importante cantidad anual a la *Fundación X Prize*, cuyo fin es premiar los avances científicos que ayuden a la humanidad.

Finalmente, Garriot es también vicepresidente de *Space Adventures*, la empresa líder mundial en turismo espacial. Ésta es la empresa que ha enviado hasta el momento a 5 turistas espaciales a la *Estación Espacial Internacional*, y Richard Garriot será el sexto visitante privado de la misma en Octubre de este año, convirtiéndose en el segundo hijo de un astronauta que a su vez visita el espacio.

Sin duda se trata de uno de los personajes más grandes surgido del mundo de los videojuegos. Esperemos que nos traiga alguna nueva idea genial de su viaje interplanetario.





POR
PABLO VILCHES

RARE, RELATO DE UN CUENTO



Rare es una de las compañías más aclamadas de la última década, y probablemente, el estudio de desarrollo más talentoso que ha dado Europa. Los hermanos Stamper son los responsables de haber dado forma a una empresa que actualmente ofrece demasiadas incógnitas. ¿A qué se debe esta valoración? ¿Prensa y público están siendo injustos con *Rare*? Hablemos un poco sobre estos chicos de Reino Unido.

En el año 1982 los hermanos Tim y Chris Stamper fundan *Ashby Computers and Graphics Ltd.*, aunque publicarán con el nombre de *Ultimate Play the Game*. Es un año muy importante para la industria, naciendo en el mismo año estudios como *Electronic Arts* y *Lucasfilm Games*. *Ultimate* publicó principalmente para los ordenadores *Sincalir ZX Spectrum*, aunque también en *Commodore 64*, e incluso *BBC Micro*. Sus juegos seguían las pautas que parecían estar marcadas en la mayoría de los juegos de *Spectrum*, ofreciendo una jugabilidad *arcade* muy en la línea del desarrollo ofrecido en Reino Unido (en los Estados Unidos de América, por ejemplo, el *Apple II*, comparable al fenómeno *Spectrum* en Europa, ofreció un modelo distinto de juego). El resultado fue un buen puñado de éxitos como *Jetpac* (sólo la versión de Spectrum vendió 300.000 copias), *Lunar Jetman* (secuela de Jetpac), *Sabre Wulf* (350.000 copias sólo en Spectrum), *Underwurlde* o *Knight Lore* (toda una revolución). Esto tenía un precio personal, ya que los hermanos Stamper dicen haber trabajado en esa época 80 horas a la semana durante siete días, tomándose por navidad un par de días libres. También obtuvieron grandes beneficios económicos con el que permitirse lujos como la compra de un *Lamborghini*. En definitiva, se convirtieron en el estudio más importante de Reino Unido, y probablemente de Europa.



En medio de todo este éxito ocurrió algo que nadie esperaba, un movimiento incomprensible, una decisión a la altura de las actuaciones de todos esos genios cuyas inclinaciones fueron cuestionadas en un momento concreto para que sólo el tiempo les diese la razón. En el año 1984 los Stamper venden la marca *Ultimate*, y deciden abandonar al *Spectrum*. El motivo es el descubrimiento de una máquina de *Nintendo* llamada *Famicom*, disponible en Japón, y que iba a ser comercializada (bajo el nombre *NES*) en los Estados Unidos. Como no disponían de ningún kit de desarrollo, decidieron comprar una *Famicom* y todo el software disponible para ella. ¿El objetivo? Comprender el funcionamiento del sistema a través de la ingeniería inversa. Para este propósito crearon una subdivisión secreta llamada *Rare*, cuando aún publicaban bajo la marca de *Ultimate*, la cual se encargaría de llevar esta investigación en la máquina de *Nintendo*. Ya había nacido *Rare*.

Para el año 1985 habían hackeado el sistema por completo, así como empezado a desarrollar software para ella, siendo los primeros occidentales en programar "algo" para la *Famicom*. Mostraron parte del trabajo a *Nintendo*, y estos, tras quedarse impresionados con lo que habían hecho los hermanos ingleses, decidieron enviarles documentación oficial, pero a los Stamper ya no les hacía falta. Sólo necesitaban una cosa: la licencia oficial de desarrollo de *Nintendo*, cosa que por supuesto, también habían conseguido. Pero la cosa no se quedó ahí. *Nintendo* en aquella época establecía una serie de condiciones muy exigentes para cumplir con su "Sello de Calidad de Nintendo". Entre esas condiciones, sólo se permitía publicar un número limitado de juegos al año, pero a *Rare* no se le impusieron estas limitaciones. Entre el año 1987 y 1991 *Rare* lanzó al mercado el espectacular número de 44 títulos (estáis leyendo bien, 42 en la *NES*, y 2 en la *Game Boy*). Ninguno de ellos fue un exitazo, pero eran rentables, y posicionaban económicamente al estudio para afrontar los costosos desarrollos que la industria debería afrontar en el futuro. En *Rare*, nada se hacía al azar, mostrando una y otra vez una visión de futuro que sólo ahora somos capaces de entender. Muchas empresas del gremio comentaron en alguna ocasión que estaban locos.

Los medios y el público inglés no entendían que es lo que estaba haciendo *Rare*, su joya en bruto. Abandonaban el *Spectrum* para pasarse a una máquina de una desconocida compañía japonesa de la que nadie había oído hablar en la vida. Pero los hermanos Stamper, firmes en su visión, respondieron que el *Spectrum* estaba muerto, y que *Nintendo* había ideado el modelo de futuro para los videojuegos. Aún con los números en la mano (10 millones de *Famicom* vendidas en Japón, y un éxito de ventas en el reciente lanzamiento americano), su país no parecía hacerles caso. Tim y Chris Stamper amaban los videojuegos japoneses, y de hecho, bebieron mucho de ellos. Es más, alabaron la figura de *Nintendo* en el Reino Unido, aún a costa de quitarse méritos. Lo único que querían es que sus compatriotas entendiesen lo que ellos parecía que habían entendido.

Nos colocamos en el año 1992, y *Rare* se ve con una situación económica bastante interesante. Ya va siendo hora de invertir este dinero en algo serio de verdad, y no, no fue un videojuego. La inversión consistió en la estación de trabajo *Silicon Graphics* más importante de todo el Reino Unido. El objetivo era utilizar la rica paleta de colores de la recién lanzada *Super Nintendo* para combinarla con espectaculares escenarios renderizados, dando lugar a un juego de plataformas con aspecto 3D. La idea gustó en Kyoto, y decidieron ofrecerle a *Rare* la oportunidad de trabajar con *Donkey Kong*. *Rare* aceptó, y paralelamente trabajó en un juego de lucha: *Killer Instinct*. En el año 1994 *Donkey Kong Country* es lanzado al mercado, llegando a vender 8.5 millones de unidades y obteniendo numerosos premios. Es entonces cuando *Nintendo* decide comprar el 25% de las acciones de *Rare* (que más tarde se convertirían en el 49% de las mismas). *Rare* se convierte en *second party* de *Nintendo*, y da comienzo la etapa más brillante y mágica de este estudio. También cambia la política del estudio, dejando de lado esos tiempos en los que podían llegar a lanzar 17 títulos en un año. Ahora lo que primaba era lanzar títulos con un razonable período de desarrollo que asegurase una alta calidad.

Super Nintendo y *Nintendo 64* elevaron a *Rare* al cielo. Habían pasado a ser sinónimo de gran calidad, gran jugabilidad, y grandes gráficos. Durante estos



años podemos destacar éxitos en *Super Nintendo* como *Donkey Kong Country*, *Donkey Kong Country 2*, *Donkey Kong Country 3*, o *Killer Instinct*. Y a partir de 1997, en *Nintendo 64*, podemos contabilizar nombres como *Killer Instinct Gold*, *Diddy Kong Racing*, *Blast Corps*, *GoldenEye*, *Donkey Kong 64*, *Banjo Kazooie*, *Banjo Tooie*, *Conker's Bad Fur Day*, y *Perfect Dark*. También lanzaron juegos y ports (principalmente *ports*) para las portátiles de *Nintendo* (*Game Boy* y *Game Boy Color*), e incluso algún juego en la *Megadrive* de *SEGA* antes de ser *second party* de *Nintendo*. De entre estos, merece la pena hablar del *GoldenEye*, juego ambientado en la película de *James Bond*, el agente *007*, y que, a pesar de lo que muchos puedan pensar, no fue cosa de *Rare*, sino de *Nintendo*, puesto que a los Stamper no le gustaba nada la idea. Aun así accedieron, pero poniendo al frente del desarrollo a programadores sin experiencia o con muy poca. No era un juego importante ni para *Rare* ni para *Nintendo*, y así se pudo ver con el escaso seguimiento que tuvo en las ferias. Cuando el juego fue lanzado cosechó críticas excelentes por parte de la prensa y público, y aunque no fue un lanzamiento superventas, finalmente acabó vendiendo ocho millones de copias al final del ciclo de vida de la *Nintendo 64*. Hoy en día, programadores que participaron en ese desarrollo y que aun siguen en *Rare*, no terminan de entender la calidad final del juego dado el poco entusiasmo que hubo en el proyecto.

A partir de 1997 empieza la fuga de desarrolladores a otras empresas. *Rare* sufrirá una lenta sangría hasta pasar a formar parte de *Microsoft*, y aun así, con *Microsoft* afrontará su mayor pérdida como veremos más adelante. El caso más conocido es el de David Doak, escritor y diseñador de *GoldenEye* que, junto con otros miembros de dicho proyecto formaron la conocida *Free Radical*. Martin Hollis, encargado del desarrollo de *GoldenEye*, pasó a formar parte de *Nintendo of America*, con el objetivo de asesorar el desarrollo de *Gamecube* (más adelante fundaría *Zoonami*). Son sólo ejemplos sonados, pero en el año 1999 la empresa llegó a perder a 50 trabajadores.



Rare no vendía mal (aunque no tan bien como se imagina la gente), sus juegos tenían buena crítica, pero de una manera poco apreciable empezaba a surgir un sentimiento de desconfianza en las comunidades de nintenderos. Juegos hoy en día muy bien valorados como los *Banjo* o el *Donkey Kong 64*, fueron en su momento fuertemente criticados en algunos sectores, puede que debido a la larga sombra de *Super Mario 64*, pero la crítica estaba ahí, y es que los fans de *Rare* seguramente esperaban alguna genialidad, y no juegos de plataformas que no llegaban al nivel del universo 3D del fontanero de *Nintendo*. Peor aun, *Jet Force Gemini* fue lapidado. Y la traca final, *Conker's Bad Fur Day*, el "simpático" juego de ardilla, no sentó nada bien en *Nintendo*, debido a su alto contenido de violencia y escatología en un mundo aparentemente inocente. Por culpa de esto, el juego no salió de Estado Unidos y Reino Unido, además de ser concienzudamente mal distribuido por *Nintendo*. El juego, que fue aclamado por el público y la prensa, no gozó de la difusión necesaria debido a una serie de políticas que *Nintendo* sostenía en aquella época. Esto tensó las relaciones, pero en absoluto fue un motivo para la separación que vendría más adelante. En el año 2002 *Rare* lanza un juego muy poco inspirado, *Starfox Adventures*. No era un mal juego, de hecho era muy bueno, pero a sus espaldas arrastraba una cantidad de promesas de las cuales casi ninguna fue cumplida. Se habló de un mundo enorme para explorar, de una experiencia de juego que podría ofrecer al usuario 100 horas de juego, de un ecosistema que parecería vivo. Al final, el juego se quedó en un clon de la saga *Zelda* de unas 20 horas aproximadas de duración, con los personajes de *Starfox* metidos con calzador en el último momento. De *Dinosour's Planet* solo quedó el espectacular aspecto gráfico.

Y sin cambiar de fecha, es ahora cuando entra en escena *Microsoft*. Los chicos de la división de *Xbox* necesitaban franquicias con renombre de manera urgente, y *Rare* tenía muchas, además de aportar un estilo de juego muy personal. El rumor saltó a la red, y realmente, poca gente se lo creyó. Nada hacía pensar que *Nintendo* fuese a permitir que un estudio tan importante, y que tanto había aportado a *Nintendo* a pesar de las dificultades puntuales, se dejase escapar. Pero los hermanos Stamper lo tenían claro, y vendieron el 51% de las acciones de la empresa a *Microsoft*, sabiendo que *Nintendo* vendería a *Redmond* el 49% de las acciones restantes. Así fue, *Nintendo* vendió su parte por 377 millones de dólares, y dicen las malas lenguas, que no fue muy entusiasta a la hora de detener la operación entre los Stamper y Bill Gates, y no es de extrañar, ya que en Kyoto se permitió que *Rare* se quedase con todas las franquicias que habían creado, incluyendo juegos en desarrollo para *Gamecube* (es el caso de *Kameo* y *Perfect Dark Zero*). Tras esto, 30 empleados abandonaron la empresa.

Microsoft había dado un golpe de efecto muy importante, y empezaba a consolidar su posición en la industria. En *Redmond* se dijo: "*Rare* es la pieza central en nuestra estrategia de videojuegos". Declaración que no es de extrañar, ya que al fin y al cabo, *Rare* seguía siendo una de las compañías más respetadas de occidente. En el otro lado, la comunidad nintendera estaba en estado de *shock*. De acuerdo, últimamente se les había criticado por haber perdido parte de esa magia, pero pasarse al bando enemigo era inadmisible. Aquí empezó una campaña de desprestigio hacia *Rare* hervida directamente en los foros de las comunidades de seguidores de *Nintendo*: "*Rare* ya no es una gran empresa", "*Rare* ya no tiene entre sus filas a mentes destacables", y por supuesto, "*Nintendo* ha vendido a *Rare* porque sabe que el estudio va a caer en picado". Por si esto no fuera poco, nuestros amigos de Reino Unido no tenían nada mejor que hacer que lanzar al mercado el infumable *Grabbed by the Ghoulies*, y un remake del *Conker's Bad Fur Day* que llegaba en un momento que no era para nada adecuado.

Hoy en día, y con la *Xbox 360* en el mercado, parece que *Rare* recupera poco a poco su credibilidad. Si bien *Perfect Dark Zero* y *Kameo* no fueron exitazos, sí que cumplieron las expectativas de ventas, y en el caso de *Kameo* se demostró que *Rare* todavía era capaz de ofrecer juegos mágicos. Con *Viva Piñata* ofrecieron otro juego notable y diferente que, aunque no vendió más de 500.000 unidades, ha servido para establecer otra franquicia. Gregg Mayles, diseñador jefe de *Viva Piñata*, ha culpado a *Microsoft* por las escasas ventas del producto, considerando que se destinaron muchos recursos para promocionar *Gears of War* en perjuicio de la promoción de *Viva Piñata*.

En 2007 los hermanos Stamper decidieron abandonar *Rare* para embarcarse en otros proyectos personales. Dejaron atrás una empresa que ha sido una parte más que importante de su vida. Los motivos son desconocidos, ya que tan sólo declararon que era hora de hacer otras cosas. La noticia no tuvo mucha difusión, aunque suponemos que el hecho de que *Rare* ya no les pertenecía pudo tener algo que ver. ¿Se arrepienten de la decisión que tomaron al decidir dar a *Microsoft* la totalidad de sus acciones? Por ahora no lo sabemos. Quizá, si finalmente continúan en el mundillo de los videojuegos, les veamos programando otra vez para *Nintendo*. También puede que hayan decidido jubilarse, dado el posible patrimonio económico que tendrán.

Respecto al estudio, veremos qué nos depara el futuro. Parece que *Banjo 3* está en desarrollo, y se rumorea que un *Kameo 2*, un *Perfect Dark 2*, o un nuevo juego podría estar gestándose en la cocina. El estudio no está en malas manos. Gregg Mayles ha sustituido a Tim y Chris Stamper, y probablemente sea el mejor sustituto. Ha demostrado entender esa visión particular que *Rare* tiene de los videojuegos, y eso es lo más importante. *Microsoft* mantiene su confianza en ellos, y ¡como para no mantenerla después de los 377 millones de dólares gastados!

Para acabar, y como no podría ser de otra manera si hablamos de *Rare*, aquí acaba este cuento, o mejor aun: "Y colorín colorado, este cuento no ha acabado..."



POR
ESTER SÁNCHEZ

TUTORIAL: PROGRAMACIÓN DE JUEGOS PARA GP2X



Introducción

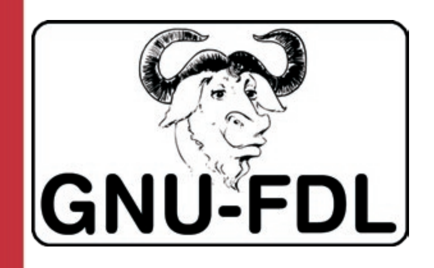
Dado que normalmente no se aprende nada sobre programación gráfica en las universidades/módulos/academias... (salvo excepciones), siempre cuesta un poco de trabajo saber cómo va “esto de programar juegos”. La pregunta que alguna vez te habrás hecho, al igual que yo y muchos otros, imagino que habrá sido por dónde o cómo empezar.

El objetivo de este mini tutorial es dar respuesta a esa pregunta y a otras que puedan salir al principio. Me gustaría que esto sirviera para dar un “empujoncito” a aquellos que no se han atrevido a entrar en el desarrollo de juegos por navegar en un mar de dudas.

¿Por qué GP2X + C + SDL y no otras opciones? Pues he elegido la *GP2X* porque es la consola que, para mí, da más facilidades a la hora de programar (aunque la Nintendo DS se le acerca peligrosamente). Es cómoda puesto que podemos probar las cosas directamente en la consola conectándola por USB al PC y entrando por *TELNET*, a parte de la opción tradicional de meter y sacar la tarjeta SD las veces que haga falta.

Como vimos en un número anterior de la revista, existen varias opciones a la hora de programar para la *GP2X*. Mi elección de SDL y C se debe sencillamente a que es la opción que mejor me manejo. Además, la ventaja del SDL es que podemos probar todo lo que hagamos en el PC, por lo que nos evitamos estar probando en la consola constantemente.

Antes de “meternos en faena” he de decir que parto de la base de que tenéis conocimientos de programación. No necesariamente de C, pero sí de cosas como qué es una variable, un bucle, etc. Si no los tenéis hay un montón de tutoriales por la red sobre el tema, concretamente en www.lawebdelprogramador.com.



¿Qué necesito para empezar?

Como seguramente hayáis adivinado (es bastante obvio) necesitamos el compilador y las librerías para poder ponernos en marcha.

El micro que lleva la *GP2X* tiene arquitectura ARM, lo que significa que si creamos un ejecutable en nuestro PC (arquitectura x86 probablemente), no nos servirá. Pero no hay problema, porque para ello existen los compiladores cruzados. Esto significa que necesitamos un compilador "especial" que cree ejecutables válidos para la consola. Lo mismo nos ocurrirá con las librerías, han de estar compiladas específicamente para la *GP2X*.

¿Qué dónde lo puedes conseguir? Pues dependiendo del sistema operativo de tu ordenador tienes varias opciones:

- Windows: Podéis descargaros el SDK oficial (compilador + librerías) que incluye además el IDE DevC++ de: http://dev.gp2x.com/sdk/gp2xsdk_windows.zip. Para instalarlo es descomprimir y listo.

- Linux: Aquí tenemos dos opciones principalmente, bien descargarnos el SDK oficial, bien uno compilado por alguien que puede que incluya más opciones que el primero. Mis recomendaciones son:

Si tu elección es el SDK oficial: http://dev.gp2x.com/sdk/gp2xsdk_linux.tar.gz.

Si tu elección es un SDK alternativo: http://www.misato.es/downloads/toolchain-uncanny.tgz (compilado y preparado por Uncanny -¡gracias!-). La instalación es como en windows, descomprimir y listo.

- Mac OS X Leopard: Nuestro amigo Kounch (de GP32spain.com) ha preparado un tutorial de cómo instalar el SDK en Leopard, además de hacer unas plantillas para Xcode 3. Podéis encontrarlo aquí: http://www.gp32spain.com/foros/showthread.php?t=51307.

Además, sería recomendable bajar las librerías a tu ordenador (tanto las normales como las de desarrollo) para poder hacer pruebas. Se pueden conseguir en la web oficial de SDL: www.libsdl.org

¿Por dónde empezar? Cosas a tener en cuenta

Programar un juego difiere un poco de programar cualquier otra aplicación (sobre todo si es de consola en vez de una aplicación gráfica).

En un juego los gráficos son importantes, al igual que la música, por lo que no es sólo picar código, sino que también tenemos que saber mostrar cosas por pantalla, sacar sonido por los altavoces, etc. Para esas tareas tenemos a las SDL, que nos harán el "trabajo sucio" mucho más fácil.

Hay varios "conceptos" básicos a tener en cuenta en la programación de videojuegos que pasaré a resumir:

1) Estructura de un juego

Un "esqueleto" de cómo sería un juego puede ser este:

```
int main(){
 INICIALIZAR_VIDEO_Y_SONIDO();
 INIT_Joystick();
 done = 0;
 while(!done){
    Mostrar_graficos();
        .
        .
        .
    //resto de código
    Capturar_eventos_de_joystick();
    Procesar_esos_eventos();
    if (gameOver){
       done=1;
    }
  }
}
```

Me he inventado las funciones, por supuesto, y no tienen por que ir en ese orden. Pero lo importante es el



while(!done). Este bucle casi infinito (se ejecutará hasta que acabemos la partida) es en el que se hace todo. Normalmente es fuera donde podemos poner nuestro menú del inicio del juego, o inicializaciones varias (como se ve en el esquema), y dentro de él mostramos las diferentes pantallas, hacemos el movimiento de los personajes, etc. En algunas consolas (como la *NDS* por ejemplo) que no tienen un menú al inicio, sino que simplemente carga el juego según enciendes, ese bucle podría cambiarse por *while(1)*. Sin embargo en la *GP2X* tenemos la posibilidad de salir al menú, de modo que debajo de eso podríamos tener el código de esa salida.

2) Mostrar imágenes

SDL lleva unas funciones para las imágenes bastante completas. Eso sí, limitadas para cargar bmp. Si queremos cargar imágenes en otros formatos, necesitamos ayudarnos de *SDL_image*. De todos modos, para nuestro tutorial básico usaremos sólo SDL sin apoyarnos en *SDL_image*, puesto que con un bmp nos valdrá para entender el funcionamiento. Para mi ejemplo he tomado prestado unos *sprites* de *Super Mario Bros*. Mostrar una imagen es tan sencillo como:

```
SDL_Surface *imagen = SDL_LoadBMP
("imagen.bmp");
```

SDL_Surface es una estructura de SDL para el manejo de imágenes principalmente (también para la pantalla, por ejemplo). Esta estructura nos proporciona información sobre su anchura, altura y formato de píxel, entre otras cosas. Con esa línea de código ya tendríamos cargada en memoria nuestra imagen, pero ahora debemos mostrarla. Para ello nos apoyaremos en otra estructura de SDL que es el *SDL_Rect*. Con ella indicaremos dónde vamos a colocar la imagen, dándole coordenadas, incluso podríamos definir una parte de la imagen para que sólo mostrase eso. Digamos que *SDL_Rect* es un "rectángulo" donde va a ir la imagen o parte de ésta en la pantalla. Mostrar la imagen en la pantalla ser haría con:

```
SDL_Rect destino;
//le damos valores a destino:

destino.x = 0;          //coordenada X de la pantalla
destino.y = 0;          //coordenada Y de la pantalla
destino.h = imagen->h;  //altura de la imagen
destino.w = imagen->w;  //anchura de la imagen
```

Con esto, hemos preparado el "rectángulo" donde vamos a mostrar la imagen. En este caso, se mostrará en el punto (0,0), osea en la esquina superior izquierda de la pantalla.

```
SDL_BlitSurface(imagen, NULL, screen,
&destino);
```

Con esta función le indicamos que copie la imagen (*imagen*) en la pantalla (*screen*) en el rectángulo que hemos definido (&destino). El segundo parámetro puede ser un *SDL_Rect* que defina una parte de la imagen, si se deja en *NULL* copia la imagen entera. Así tendríamos cargada la imagen en su posición y lista para pasarse a la pantalla, que se hace con la función:

```
SDL_Flip(screen);
```

No voy a explicar mucho más de las imágenes salvo una cosa que creo que sí es importante: las transparencias. Si tenemos una imagen de un personaje, que tiene forma de rectángulo, pero el personaje ocupa menos que la imagen, se mostrará nuestro personaje con un color alrededor (rellenando ese rectángulo). ¡Y nosotros sólo queremos que se vea nuestro personaje! Eso se puede solucionar con las transparencias. Básicamente consiste en definir un color que no se pintará en pantalla. (Utiliza uno "raro" que no tenga ninguna de tus imágenes, por ejemplo el rosa fosforito). Las transparencias se definen en SDL con:

```
SDL_SetColorKey(imagen, SDL_
SRCCOLORKEY, SDL_MapRGB(imagen-
>format,255,0,255));
```

El primer parámetro es la imagen que tendrá transparencias. El segundo, es una constante que indica que se trata de una transparencia y por tanto no tiene que pintar los píxeles de ese color. Por último el tercer parámetro indica un color en RGB (255, 0, 255) que será el transparente.

Hasta aquí la introducción sobre las imágenes. Hay bastante más sobre el tema pero prefiero dejarlo para otra ocasión. Con esto ya podemos mostrar a nuestro personaje en pantalla sin ningún problema.

3) Particularidades de SDL para *GP2X*

Las SDL tienen muchos modos de funcionamiento. Dado que nuestro caso es una consola con menos recursos que un PC, debemos tener en cuenta ciertas restricciones. Esto sí es exclusivo para *GP2X*, y está sacado de su *wiki* (wiki.gp2x.org):

- Modo de Vídeo:

```
SDL_SetVideoMode (320, 240, 16, SDL_SWSURFACE);
```

Ese es el modo de vídeo que funciona perfectamente. 320x240 a 16bpp (aquí nos vale también 8) y *SDL_SWSURFACE*. No funciona el doble *bufer*, por lo que no intentéis inicializaciones del tipo: *SDL_DOUBLEBUF|SDL_SWSURFACE*. Si no entendéis muy bien qué es cada parámetro, no os preocupéis. Lo importante es que con esta función inicializamos el modo de pantalla (320x240 de resolución, a 16 bits).

- Mapeo del Joystick:

El joystick y los botones en SDL se controlan asignándole un entero a cada pulsación o botón. Para la *GP2X* esto corresponde a los siguientes números:

Joystick	Valor	Botón	Valor
Arriba	0	A	12
Abajo	4	B	13
Izquierda	2	Y	14
Derecha	6	X	15
Arriba-Izquierda	1	R	10
Arriba-Derecha	7	L	11
Abajo-Izquierda	3	Start	8
Abajo-Derecha	5	Select	9
Stick "Click" (solo F100)	18	Vol-	17
---	---	Vol+	16

Su utilización es muy sencilla. Por ejemplo, si definimos el botón A y el B así:

```
#define Boton_A  (12)
#define Boton_B  (13)
```

Para saber cuál hemos pulsado habría mirar en la cola de eventos si el valor del evento corresponde con el número del botón:

```
while(SDL_PollEvent(&event)) //mientras haya eventos en la cola
while(SDL_PollEvent(&event)) //mientras haya eventos en la cola
{
        switch(event.type) // miramos el tipo del evento
        {
                case SDL_JOYBUTTONDOWN: //si el evento es que se ha pulsado un botón
                        switch(event.jbutton.button) //miramos qué boton es
                        {
                                case Boton_A: printf("Has pulsado la A\n");break;
                                case Boton_B: printf("Has pulsado la B\n");break;
                        }
                break;
        }
}
```

Yo aconsejo crear un *.h* con todas las definiciones de botones e incluirlo en el *main.c* (o donde se vaya a usar). Además, este os puede valer para el resto de aplicaciones que hagáis para la consola.

- Ocultar el cursor

Si hacéis pruebas os encontraréis un cursor en la esquina superior izquierda. Para evitar que esto salga, y a menos que vayamos a usar un cursor en nuestro juego, usaremos esta línea después de inicializar el vídeo:

```
SDL_ShowCursor(0);
```



- Cómo salir al menú de la consola

Como hemos visto, podemos salir al menú de la consola una vez hayamos terminado con nuestro juego. Esto se haría así:

```
#include <unistd.h>
...
chdir("/usr/gp2x");
execl("/usr/gp2x/gp2xmenu", "/usr/gp2x/gp2xmenu", NULL);
```

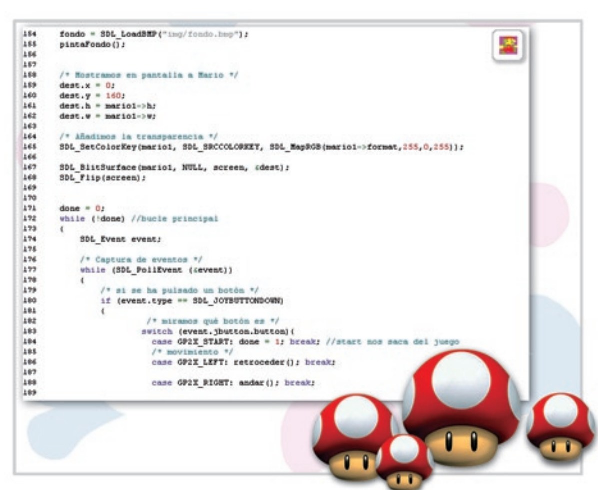
Esas son las principales cosas a tener en cuenta. Si queréis más información sobre SDL mirad la web oficial o el wiki de *GP2X*.

4) Ejemplo de código

Puedes descargarte un ejemplo de código de mi página web: www.misato.es/ejemploMario.zip. En el zip viene incluido el código fuente, las imágenes (en su carpeta correspondiente) y el ejecutable de la GP2X por si quieres probarlo inmediatamente. En este ejemplo se muestra un Mario que anda o retrocede según pulsemos el *joystick* hacia delante o hacia atrás. Para salir, pulsar *START*. Vamos a comentar un par de cosas sobre él:

- El pintado de las imágenes:

Siempre se repinta el fondo y después la imagen que queramos poner (en este caso la imagen que corresponda al Mario andando). Esto es porque tienes que verlo como "capas" que se van superponiendo. Es decir, el fondo al ser lo que ocupa toda la pantalla se debe pintar primero porque, si se pintara después del Mario, le taparía por completo dadas sus dimensiones. Por tanto, si queremos cambiar la imagen de Mario, hay que "tapar" lo anterior (si no se nos repetiría el Mario) y ¿qué mejor manera de "taparlo" que repintando la pantalla? Esto no tiene por qué ser así siempre, a veces si sólo queremos actualizar una parte de la pantalla en concreto podemos sólo repintar ese trozo. Esto optimiza bastante el código y aumenta la velocidad, pero eso es una técnica más avanzada que no tiene cabida en un tutorial básico como éste.



```
fondo = SDL_LoadBMP("img/fondo.bmp");
pintaFondo();


/* Mostramos en pantalla a Mario */
dest.x = 0;
dest.y = 160;
dest.h = mario1->h;
dest.w = mario1->w;

/* Añadimos la transparencia */
SDL_SetColorKey(mario1, SDL_SRCCOLORKEY, SDL_MapRGB(mario1->format,255,0,255));

SDL_BlitSurface(mario1, NULL, screen, &dest);
SDL_Flip(screen);


done = 0;
while (!done) //bucle principal
{
    SDL_Event event;

    /* Captura de eventos */
    while (SDL_PollEvent (&event))
    {
        /* si se ha pulsado un botón */
        if (event.type == SDL_JOYBUTTONDOWN)
        {
                /* miramos qué botón es */
               switch (event.jbutton.button){
                 case GP2X_START: done = 1; break; //start nos saca del juego
                 /* movimiento */
                 case GP2X_LEFT: retroceder(); break;

                 case GP2X_RIGHT: andar(); break;

```

- La captura de eventos:

En el código sólo hay capturados los eventos de *joystick* de la *GP2X*. Pero podría incluirse también la captura de eventos de teclado, por si se prueba en el PC. Esto sería simplemente comprobar que el tipo de evento es el de una tecla pulsada *(if (event.type == SDL_KEYDOWN))* y ver qué tecla es en concreto *(switch (even.keysym.sym){ case SDLK_A: /*lo que sea */break;)}*, al igual que hacemos con los botones de la *GP2X*. Como ves, manejar eventos no es complicado, y si le echas un vistazo a la documentación de SDL, verás que hay muchos tipos de ellos, desde los que hemos visto en el programa de ejemplo, hasta eventos que capturan si has cambiado el tamaño de la ventana, por ejemplo. Y en todos los casos el manejo es así de sencillo..



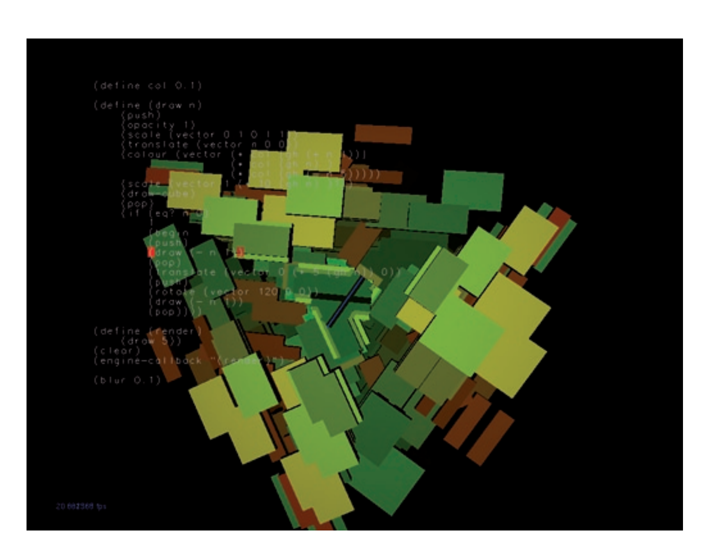
Y hasta aquí este pequeño tutorial. Si no ha quedado algo claro puedes enviarme tus dudas a ester@altmagazine.net y las contestaré cuanto antes.

Espero que haya servido para dar un empujoncito a quien quiera empezar a hacer sus cositas en la GP2X y ¡espero que podamos verlas pronto!

FLUXUS: EL NOBLE ARTE DE FLUIR



Cuando le comenté a un amigo (de los que no te hacen concesiones) que estaba aprendiendo a usar un programa que te permitía crear visuales por medio de un código me hizo dos preguntas “¿qué?” y “¿para qué?”. La primera es de fácil respuesta, especialmente para los que tenemos asumida nuestra rareza, y se responde con una obviedad del palo de “una excentricidad” (nótese de ahora en adelante mis reservas a la utilización en este contexto de la tan traída palabra *friki*, herramienta utilísima en la tarea de normalización social y consiguiente decadencia cultural). La segunda pregunta de mi amigo tiene más intríngulis, ese “¿para qué?” maldito. Pues para nada. Eso es lo más maravilloso del asunto. Los más modernos del lugar dirán que tiene gracia sacar la imagen a un proyector por ejemplo y usarla para amenizar sus fiestas de “machacapastis” o algo así. Pues sí, no está mal pensado, pero eso ya entra dentro del juego de la triste mente occidental de tratar de producir algo útil a toda costa dejando como anécdota lo importante, el hacer algo por el mero hecho de hacerlo.



Alguien podría decir que eso es arte, otros dirán que la programación en sí misma ya es arte y algunos no habrán llegado siquiera a leer hasta aquí. Para éstos hay un futuro hermoso de felicidad esperándoles, con mujer rubia y servicial, casa a las afueras, coche deportivo, valla blanca, jardín y perro. Para el resto sólo lucha, lucha, lucha y *fluxus*.

Ya de antemano el programita te pone a prueba.



Los autores quieren saber si estás realmente loco o es sólo una pose infantil. Para ello han usado las librerías más absurdas que han sido capaces de encontrar en sus versiones más inverosímiles. Esto hace que sea casi imposible instalar correctamente todo el tinglado, por lo menos en el sistema ese del pingüino y los comunistas. Probablemente en *Mac* todo irá mejor y más fácil porque todo el mundo sabe que en los *macs* la informática es como un anuncio de compresas.

Si has sido capaz de configurar, compilar e instalar los hasta siete paquetes de dependencias exóticas de los que hace gala el asunto, podrás probar lo que es el verdadero *livecoding*, esto es, picar código y ver los resultados inmediatamente en pantalla.

Realmente *fluxus* es un entorno gráfico que entiende el lenguaje Scheme, se puede usar para crear pequeños juegos porque admite entrada de datos por teclado, ratón y *joystick*, puede leer una pista de audio como recurso para realizar animaciones incluso por red y dispone de un pequeño motor para simular en tiempo real dinámicas de cuerpos sólidos.

La interfaz donde se escriben los pequeños *scripts* se llama *scratchpad* y está compuesta por *buffers* de texto que funcionan como las consolas de *Linux*, se puede alternar entre una y otra pulsando *CTRL* y un número del 1 al 9. Con *CTRL+0* traemos la consola del programa, donde se vuelcan mensajes internos y errores. Lo más habitual es usar un sólo *buffer* e ir ejecutando el *script* completo con F5 o una parte del mismo seleccionando un trozo del texto antes de ejecutarlo.

Se pueden crear objetos básicos como cubos con la facilidad con la que se construyen rascacielos en plena costa mediterránea, es decir, con el comando *build-cube*. Si ahora ejecutamos el programa nos aparecerá un cubo detrás del código que podremos ver desde diferentes puntos de vista usando el ratón: con el botón izquierdo rotamos la cámara, con el de en medio movemos el eje y el derecho hace un zoom.

Vamos a darle un poco de vidilla al cubo. Podemos usar el audio de ese “emepetrés” de Camela que tanto te emociona y que te da vergüenza poner cuando llegan tus amigos a casa, para hacer que nuestra escena mole de verdad. Para ello se usa un puerto de salida del servidor de audio de baja latencia *JACK*, controlado por una interfaz del tipo de *qjackctrl* y se le dice a nuestro *script* que escuche: *(start-audio “puerto-de-JACK” “bitrate” “frecuencia”)*. Creamos una función con que se repetirá en cada frame que se dedicará a cambiar el color del cubo dependiendo de ciertos harmónicos del audio con:

```
(define (render)
        (colour (vector (gh 1) (gh
2) (gh 3)))
         (draw-cube))
```

y se le dice que en cada frame se llame a la función que acabamos de crear:

```
(every-frame (render))
```

y ya está, tu cubo bailando con la voz de “chiquiprecio” y el organillo de Camela, ¿no es maravilloso?

Se me está ocurriendo que en vez del audio de un mp3 podríamos hacer la cosa mucho más dinámica, por ejemplo recibiendo la señal de línea proveniente de un teclado o una guitarra eléctrica pasada por un puerto de *JACK*. O incluso conectarle un “*sinte*” como *Spiral Modular Synth*. Un mundo de posibilidades, oiga.

En el ejemplo anterior hemos usado *colour()* para cambiar el color del cubo, y le hemos indicado que los vectores a usar (del tipo RGB) los obtenga del audio con el comando *gh*. Es una de las muchas funciones preestablecidas que entiende el intérprete. Con *(scale(vector “x” “y” “z”))* cambiamos el tamaño del objeto, con *(translate (vector “x” “y” “z”))* lo situamos en el punto que queramos en la escena y con *(rotate (vector “x” “y” “z”))*... pues eso. Si combinamos lo que ya sabíamos antes con lo que sabemos ahora, podemos hacer que nuestro cubo cambie de tamaño y de color con la música. Ya sé que eso lo hace también tu Winamp propietario, privativo y fascista, pero recuerda que la clave es que aquí estamos para perder el tiempo y hacer algo que no sea útil. Aparte de hacerlo

nosotros con el sudor de nuestra frente y una sonrisilla de pistolero forastero, claro.

Un recurso guapo y que nos dará mucho juego es la recursividad. Una función que se llama a sí misma para crear una fila de elementos, una malla, etc. Por ejemplo, se podría crear un césped repitiendo un objeto parecido a una brizna, como un cilindro estrecho de color verde, y que se movieran de manera orgánica con la música. Para la tontería que estamos haciendo del cubo vamos a ver algo más simple:

```
(define (draw-row count)
     (cond
      ((not (zero? count))
           (draw-cube)
              (translate (vector 1.1 0
0))
             (draw-row (- count 1)))))
(every-frame (draw-row 10))
```

Esto mostraría una columna de cubos, concretamente 10. Sencillo ¿eh? es casi como escribir en inglés lo que quieres hacer. Jugando con los parámetros se pueden hacer cosas muy chulas como transformaciones en tiempo real de todo tipo.

La interfaz propia del programa está muy bien y además le da un aspecto interesante a las producciones por el hecho de tener el código mezclado con la escena final, pero si quieres, puedes usar un IDE externo como *DrScheme* para ir picando el *script*. Personalmente, no he sido capaz de configurar el IDE correctamente en *Ubuntu 7.10* para que se comunique con el *renderer*, pero hay gente que asegura que funciona y no seré yo quien les lleve la contraria. El hecho de que *Scheme* sea el lenguaje elegido por los programadores de *fluxus* tiene ventajas e inconvenientes. Puede resultar extraño, por ejemplo, que para realizar operaciones aritméticas el operador vaya antes que los datos (una multiplicación en este lenguaje sería algo como **3 4*) pero hace que exista una coherencia entre todas las llamadas a funciones ya que se invocan todas de la misma manera. Otra particularidad es la inexistencia de bucles, la única manera de hacerlos es por medio de la recursividad, y esto, como bien sabéis los programadores y otras malas gentes, es peligroso.

De manera experimental podrás encontrar extensiones para crear texturas de vídeo, renderizado por hardware, propiedades de materiales en objetos y otras cosas que no me ha sido posible probar. Por ahí he leído también que han añadido un modelador de polígonos del estilo de la tortuguita de *LOGO*, con el que podrías crear en tiempo real y en directo nuevas formas y unirlas entre ellas. Eso está también en mi lista de tareas pendientes pero suena muy bien.

Cómo gastes tu tiempo ya es cosa tuya. Si te ha despertado la curiosidad este pequeño artículo, busca en la sección de documentos de www.pawfal.org/fluxus pequeñas pistas y ayudas para hacerte con el programita. Si haces algo y no tienes amigos para compartirlo (cosa que puede pasar) házmelo saber en mi correo (cervantes@altmagazine.net). ¡Buen fluxus!



POR

IKER UNAI LARBURU

¡KARAMBA! EL DESEMBARCO DE ESCRITORIO

Muy bien reclutas, parece ser que algunos habéis aguantado la primera lección sin correr llorando a las faldas de vuestra mamá. Seguro que ahora os creeréis muy buenos, unos tipos duros, mejores que los demás. Pues, ni se os ocurra pensarlo, que ni siquiera pase por vuestra mente esta idea. Tened por seguro que no seréis buenos hasta que yo os diga que sois buenos, y de momento ninguno habéis demostrado nada, ¿entendido? ¡escoria! Pero dado que estáis aquí, y hoy me siento magnánimo, os haré el cumplido de no consideraros tan malos como la chusma típica, pero que no se os suba a la cabeza.

Bien, habiendo aclarado esto, me toca deciros que se acabaron los días de campamento. A partir de este momento estaréis en la primera línea de batalla. Si permanecéis unidos y os ayudáis unos a otros es posible que vea a alguno de vosotros en futuras misiones. Prestad atención, que ésta va a ser vuestra primera misión en solitario: una incursión en la región de la interfaz gráfica.

Sí, habéis oído bien, reclamaremos una porción del escritorio, estableceremos en ella nuestra base y desde allí lanzaremos patrullas de reconocimiento para monitorizar el sistema. Seguro que no tengo que contaros que en la sociedad de la información en la que vivimos el ordenador ha evolucionado de ser una herramienta para el trabajo a ser todo un elemento multifunción que no sólo sirve para escribir aburridos documentos o generar enormes tablas de números y cálculos entre ellos. Hoy en día un ordenador es, aparte de un equipo de ofimática, una reproductor de audio y vídeo, un álbum fotográfico, un buzón de correo, y un sinfín de utilidades más que no viene a cuenta mencionar. Es de vital importancia para nosotros tener controlado lo que pasa en

nuestra máquina, de manera que pongamos en uso los conocimientos adquiridos en el campamento anterior para alcanzar este objetivo.

¿Cuántos de vosotros habéis realizado alguna misión de reconocimiento de vuestro ordenador? ¡Oh, vamos!, seguro que alguno tiene un monitor del sistema. ¿Ninguno? Pues entonces, tal vez saquéis algo de provecho de esta misión, si conseguís volver ilesos.

Briefing

Antes de que mañana salga el sol desembarcaremos en la zona superior izquierda del escritorio (aunque os permitiré elegir otra zona si vuestro fondo de escritorio no os permite alojaros allí, ¡pero no os acostumbréis a tanta amabilidad!). Allí estableceremos la base de nuestras operaciones. Para levantar los cimientos de nuestra base, vamos a utilizar una herramienta llamada *Karamba*. Podéis obtener información sobre ella en *sourceforge* (http://karamba.sourceforge.net/). *Karamba* es una aplicación de KDE que nos permitirá mostrar información obtenida a través de distintos sensores en el escritorio e, incluso, ejecutar *scripts phyton* para interactuar con distintos elementos. Sin embargo, dado que somos comandos y no unos simples *boyscouts*, que se van a cantar canciones alrededor de una hoguera, utilizaremos las mejores herramientas a nuestra disposición, es decir, una versión de esta aplicación más moderna llamada *Superkaramba*.

Si somos afortunados y el sistema en el que vamos a desembarcar es un sistema *Linux* basado en *Debian* os bastará con un *'apt-get install superkaramba'* para instalarlo. Si no, siempre podéis bajarlo de su sitio oficial en *sourceforge.net* (http://netdragon.sourceforge.net/sinfo.html) y compilarlo por vuestra cuenta. Suerte, la vais a necesitar.

Independientemente de si habéis tomado la vía fácil o el camino de los héroes para instalar la aplicación, una vez lo hayáis conseguido podéis iniciar el *Superkaramba* ejecutando el comando *'superkaramba'*. Esta aplicación dispone de un menú muy simple que nos permitirá descargar temas de la red o cargar alguna que ya esté en nuestra máquina. Pero, ¿que es un tema? Pues muy fácil, alfeñiques. Un tema es un fichero de texto que describe la funcionalidad que el *Superkaramba* tiene que ofrecer. Existen diferentes formas de obtener temas, entre ellas la página web de *kde-look.org* (http://www.kde-look.org/index.php?xcontentmode=38) o el propio menú de la aplicación. Pero huelga decir que, aunque existen diferentes monitores que nos valdrían, nosotros vamos a levantar nuestra propia base desde los cimientos hasta el último clavo que aseguren nuestros barracones.

El campamento base

Bien, habiendo acabado con los preliminares, pasemos al plato fuerte: nuestro tema, nuestra base. Podéis observar lo que queremos desarrollar en la imagen de aquí al lado.

Como podéis ver, el tema se compone de 7 secciones (sección de información del sistema, uso de memoria, uso del sistema de ficheros, uso de la red, uso de cuentas de correo, descargas de ficheros y monitor de tiempo) que comienzan en una imagen que marca su inicio. Adicionalmente dispone de una sección no visible en la que se describe el tema.

Para evitar imprevistos cuando os encontréis sobre el terreno, os voy a instruir sobre cómo allanar el camino para facilitar la construcción del tema. Vamos a crear un directorio en el que trabajar que a partir de ahora denominaré *<directorio>*. En este directorio crearemos el fichero del tema (*monitor.theme*), un directorio donde dejar las imágenes del tema (*<directorio>/imagenes*) y un directorio donde dejar los *scripts* de *phyton* que se usarán en el tema (*<directorio>/programas*).



Antes de abandonaros a vuestra suerte y, creedme, la vais a necesitar, os explicaré los elementos básicos para levantar los cimientos de la base.

Sección 0: descripción del tema

En la sección 0 vamos a fijar los parámetros globales del tema, como el tamaño del área del tema, la fuente por defecto o la imagen del fondo. El código de esta sección es el que se presenta marcado como líneas 1-8.

```
1 ##
2 ## Sección 0:
3 ## Descripción del tema
4 ## Monitor del sistema
5 ##
6 karamba x=0 y=0 w=310 h=457 locked=false
7 image x=0 y=0 path=”imagen/fondo.png”
8 defaultfont font=”Sans” fontsize=9 color=255,255,255 shadow=0
```

Como práctica común recomiendo que inicies cada una de las secciones con un comentario. En este caso la sección de comentario son las líneas 1-5. Una línea de comentario se inicia con un carácter #, ya que *Superkaramba* ignora cualquier línea que comience con este carácter.

Primero definiremos la posición y tamaño del área del tema. El área del tema es aquel que dibujará *Superkaramba* sobre el fondo del escritorio. Esta definición se hace en la línea 6 mediante el descriptor *karamba*. Los parámetros *x* e *y* definen las coordenadas desde la esquina superior izquierda del escritorio en la que se dibujará el área del tema. Los parámetros *w* y *h* son el ancho y el alto, respectivamente, del área del tema. Y al establecer el parámetro *locked* a *false* permitimos que el tema pueda ser arrastrado por el ratón. Si quisiéramos que estuviese fijo habría que fijar *locked* a *true*.

La línea 7 define la imagen del fondo mediante el descriptor *image*. Los parámetros *x* e *y* definen las coordenadas desde la esquina superior izquierda del área del tema en la que se dibujará la imagen. El parámetro *path* contiene la ruta a la imagen a dibujar.

Dado que la línea 7 dibuja la imagen de fondo, hay que tener en cuenta dos cosas: 1) que los parámetros *x* e *y* sean 0 y 2) que el tamaño de la imagen del *path* sea del tamaño establecido por los parámetros *w* y *h* de la línea 6. De esta manera conseguiremos que la imagen del fondo cubra todo el área del tema.

La línea 8 define la fuente por defecto que el tema va a utilizar cuando genere un texto mediante el descriptor *defaultfont*. El parámetro *font* determina la fuente del texto, el parámetro *fontsize* el tamaño de la letra, el parámetro *color* el color de la letra en componentes RGB y el parámetro *shadow* los píxeles de sombreado. Mediante la línea 8 estamos estableciendo como fuente de texto por defecto una *Sans* de tamaño 9 de color blanco y sin sombreado.

¡Rompan filas!

Bien, solados, esto es todo por hoy. Mirad alrededor y quedaos con las caras de los que os rodean, pues una vez que desembarquéis en el escritorio vuestras vidas dependerán de las decisiones de esas personas. Reunios con vuestro jefe de pelotón para que os emplace en el vehículo correspondiente. Recibiréis más instrucciones una vez estéis en territorio hostil. ¡Soldados, rompan filas!



ENTREVISTA CON DAVID BRAVO









[Flashback=on] En nuestro anterior número os presentamos unas curiosas historias relacionadas con el *Copyright* y leyes afines. Tal como indicamos, habíamos entresacado esas historias de un libro llamado "Copia este libro", y os presentamos brevemente a su autor, David Bravo *[Flashback=off]*. Pues bien, en esta ocasión hemos conseguido que el mismo David, a pesar de su apretada agenda, nos respondiera unas preguntas que, creemos, os podrán interesar.

David Bravo Bueno nació en 1978 en Sevilla. Se metió en la carrera de derecho en Sevilla, y a los dos meses ya comenzó a trabajar en un bufete de abogados (como secretario, por algo se empieza). Finalmente se licenció y se especializó en derecho informático. Pero esto no fue más que el inicio de su andadura.

Es uno de los principales defensores de la propiedad intelectual, y ha participado en numerosas tertulias, charlas, debates, etc. con el fin de dar a conocer a los usuarios qué es el copyleft, hasta dónde llega el derecho a la copia privada, y otros temas relacionados. Cualquiera pensaría que escuchar a un abogado hablando sobre leyes no es el mejor plan para un sábado por la tarde. Sin embargo, David sabe expresarse de modo que la gente le entienda,





y sabe captar su atención con ese tono de humor que tanto caracteriza a los sevillanos.

En su blog (http://www.filmica.com/david_bravo/) mantiene a los usuarios informados de la actualidad de estos temas, así como de anécdotas relacionadas y los artículos que publica.

En Junio de 2005 publicó su primer libro, "Copia este libro", bajo una licencia *Creative Commons*, por lo que, además del formato impreso, se puede descargar de la red en formato pdf .

Además, desde octubre de 2006 colaboró en el programa *Noche Sin Tregua*, presentado por Dani Mateo en la *Paramount Comedy*.

Y tras esta breve introducción, damos paso a la entrevista.

¿Te van los videojuegos? ¿Cuáles son tus juegos y consolas preferidas?

World of Warcraft, Counter Strike, Praetorians, Bioshock... Hay muchos. Mi consola favorita es la Wii.

Está claro que lo que quieres es llegar al público general, y das bastantes charlas. ¿Normalmente es idea tuya dar una charla, o solo lo haces si otros te invitan a que la des?

Nunca me he propuesto para dar una charla. Es una decisión de los organizadores que yo acepto cuando mi actividad profesional me lo permite, que suele ser pocas veces.

Hay mucha gente que no entiende bien ciertos conceptos legales y necesitamos que nos hagas de intérprete. ¿Nos podrías definir en palabras comprensibles los términos "ánimo de lucro", "copia de seguridad" y "copia privada"? ¿Realmente la Ley permite una interpretación tan amplia de estos conceptos o es simplemente que algunos se aprovechan de los que no sabemos entender esas leyes?

El ánimo de lucro es el ánimo de ganar dinero con la copia que realizas. La industria pretende hacer una interpretación expansiva de ese concepto equiparándolo a cualquier ventaja, beneficio o utilidad que la copia te reporte para conseguir así que las realizadas a través de redes P2P puedan considerarse lucrativas y, por tanto, delictivas. En mi opinión, el ánimo de lucro al que se refiere el artículo 270 CP no puede interpretarse de forma amplia de tal manera que exista sólo con el hecho de que obtengas cualquier provecho, ganancia o utilidad. Hay que recordar que el Código Penal se rige por los principios de intervención mínima y de proporcionalidad.

Es cierto que existen sentencias del Tribunal Supremo que interpretan el lucro como cualquier beneficio, ventaja o utilidad que algo te reporte pero también lo es que esas sentencias a las que apelan tantas veces los juristas de la industria discográfica y cinematográfica se han dictado mayoritariamente en casos de hurtos o robos. En esos casos es obvio que no sirve de nada defenderte alegando que no tenías ánimo de lucro porque el Ferrari que has robado era para tu propio uso y no para venderlo. El coche te reporta un evidente incremento patrimonial y, por lo tanto, ese beneficio, ventaja o utilidad que te aporta el coche es en sí mismo un lucro. Ese razonamiento no puede trasladarse sin más del mundo físico a las copias. No eres más rico por realizar una copia, tu patrimonio es el mismo aunque te pases la noche descargando discos.

Esa teoría que interpreta el lucro de forma expansiva desemboca en conclusiones absurdas. De seguir este razonamiento el derecho de copia privada sería impracticable puesto que toda copia sería delictiva porque siempre te daría alguna utilidad. Del mismo modo, y continuando con esa lógica, debe concluirse que el derecho de copia privada es aquel que permite al ciudadano realizar todas aquellas copias que no le sirvan para nada, que es algo así como tener el derecho a beber aire, el derecho a curarte la enfermedad que no padeces o el derecho a abrigarte cuando hace calor.

La gran mayoría de la doctrina especializada, las pocas resoluciones que hay hasta la fecha a este

respecto e incluso la Fiscalía General del Estado, coinciden en que el ánimo de lucro ha de interpretarse restrictivamente en el sentido de “ánimo de lucro comercial”.

Por otra parte, la copia privada es la copia que una persona hace para uso privado y sin un fin lucrativo (si bien la última reforma de la Ley de Propiedad Intelectual le agrega el requisito de que la copia sea obtenida de “una fuente legal”) y la de seguridad es la copia de salvaguarda que haces de tu propio original. Toda persona puede hacer copias privadas de cualquier obra intelectual, a excepción del software, del que únicamente se permite la copia de seguridad.

En cuanto al desarrollo amateur de videojuegos (siempre que sean distribuidos de forma gratuita): ¿Es legal reutilizar gráficos recortados de otros juegos? Y en cuanto a la música, ¿se pueden usar fragmentos música que tenga derechos de autor?

No. Cualquier uso que se haga de cualquier obra, por pequeña que sea la parte que uses, necesita de la autorización de los titulares de los derechos (salvo que el fragmento utilizado lo sea con fines de comentario, análisis o juicio crítico).

¿Qué ocurre con la música clásica y el cine antiguo? ¿Hay que pagar a alguien para usarlos en un videojuego amateur?

La música clásica y el cine antiguo podrá utilizarse libremente por cualquiera si ya está en el dominio público. En España el dominio público comienza a los 70 años de la muerte del autor de la obra (se computa desde el 1 de Enero del año siguiente al fallecimiento). No obstante a las obras de los autores fallecidos antes del 7 de Diciembre de 1987, se les aplica, a ese respecto, la LPI de 1879, que fija la entrada en el dominio público a los 80 años desde el fallecimiento.

Hay que tener en cuenta, no obstante, otros derechos de propiedad intelectual afines a los de autor y que pueden recaer en la obra y seguir vigentes. Eso quiere decir que una obra que ya ha entrado en el dominio público puede tener, sin embargo, derechos por la edición realizada, si hablamos de libros, o, por ejemplo, en el caso de la música clásica, derechos derivados de su interpretación, que, estando en España, en el 90% de las ocasiones será de Luis Cobos.

Hay series, películas y música a las que no puede acceder cualquiera, por no distribuirse en nuestro país. Sin embargo, sí que se pueden encontrar en las redes P2P o en páginas web, incluso subtituladas por voluntarios en el caso de series y películas (como los llamados fansubs). ¿Hay legislación a este respecto?

Lamentablemente (y es que toda esta pugna en relación con la propiedad intelectual existe porque trae problemas reales) subtitular una película y difundir ese subtítulo públicamente (más allá del ámbito personal) es una actividad ilícita, aun sin ánimo de lucro, si no se cuenta con la autorización de los titulares de los derechos (en la práctica: pagar mucho dinero). De hecho en algunos países ya se ha demandado a varias personas que han realizado subtítulos de algunas películas de forma altruista.

Lo peor de todo esto es, como casi siempre en este tipo de casos, la hipocresía. Algunas de esas mismas productoras que señalan con el dedo a muchas personas por hacer un trabajo altruista, después descargan de Internet subtítulos de aficionados y los incluyen y venden en sus DVDs. Además, claro, de atribuirse ilegalmente su autoría. No es un rumor, sino que yo mismo he llevado casos como éste del que no puedo entrar en más detalles porque determinados acuerdos extrajudiciales así me lo impiden.

En Marzo de 2005 publicaste una carta dirigida a Jose Luis Rodríguez Zapatero (presidente del estado español) en el que exponías tus pegas respecto a la nueva Ley de la Propiedad Intelectual. Mucha gente ha criticado que la forma en la que te expresabas en esa carta no era la adecuada para dirigirte al presidente. ¿Realmente enviaste esa carta, o sólo la hiciste pública, a modo de manifiesto? ¿Has recibido alguna contestación de Zapatero? ¿La esperabas?

La carta era un manifiesto que, lógicamente, no esperaba respuesta. Por otro lado, para mí el límite a la hora de dirigirse a alguien es el respeto que se le ha



de guardar a cualquier persona. Dejado establecido ese techo mínimo, no entiendo qué diferencias puede haber a la hora de tratar a un presidente del gobierno o a mi vecino del cuarto. La gente que no estaba de acuerdo con el tono de la carta a pesar de que ésta no contenía descalificación alguna es porque consideran el humor un recurso no apto para tratar "cosas serias" y para dirigirse a "gente seria", lo que equivale a desconocer prácticamente la mitad de la reivindicación política que se ha hecho a lo largo de la historia y en todas las artes. Se trata en definitiva de personas que, como dijo Chesterton, creen que lo contrario de "divertido" es "serio", cuando en realidad, lo contrario de "divertido" sólo es "aburrido". Mi carta trataba con humor un tema serio y en mi opinión si no descalificas a quien te diriges, es un tono adecuado para cualquier persona, presidente del gobierno o no.

En plena precampaña electoral, Mariano Rajoy dijo que su partido llevará "otra política" con respecto al canon impuesto por la SGAE, alegando que "lo esencial de Internet no es la copia, sino la creatividad". ¿Qué te parece?

Me parece que hay que tener cuidado. Algunas asociaciones de internautas están apoyando ciegamente la propuesta del PP de eliminar el canon sin preguntarse qué otras propuestas hace este partido en el terreno de la propiedad intelectual.

Ninguna de las posturas de las dos fuerzas mayoritarias (PSOE o PP) pueden contentar a los que estamos a favor de un derecho de copia privada entendido de forma expansiva y en contra de un canon indiscriminado.

A pesar de que ambos partidos han planteado sus posturas de la forma más demagógica posible dando a entender que sus propuestas buscan el bien de la humanidad, lo cierto es que las propuestas del PSOE son calcadas a las que realiza SGAE y las del PP a las que realiza la industria discográfica. Los primeros están a favor del canon y de una restricción sustancial de la definición de copia privada. O lo que es lo mismo: a favor de pagar para poder hacer poco más que nada.

Los segundos, a tres meses de las elecciones, se han autoproclamado los mayores detractores del canon, con el aplauso de asociaciones de internautas incluido. Sin embargo, no aclaran cuál es su postura respecto a la copia privada ni en qué situación quedará ésta si ganan las elecciones. Tirando de hemeroteca puede verse la consideración que tiene el PP de la copia privada:

El 17 de octubre de 2005 la portavoz del PP preguntó en las Cortes al presidente de PROMUSICAE lo siguiente: "Ya que existen medidas tecnológicas de protección, ¿podemos decir que es innecesaria esta regulación de la copia privada? Por tanto, ¿es innecesario un canon que compense?". Después añadió que si efectivamente existían esos sistemas anticopia en la actualidad eso "sería perfecto" pero que si no era así, habría que aceptar el canon como medida transitoria en tanto la copia no pueda impedirse de forma eficaz.

El 30 de Noviembre de 2005, el PP justificaba en el Congreso de los Diputados la enmienda que había propuesto del artículo 31 LPI diciendo que la copia privada es distinta a "los otros límites" porque "deriva de una imposibilidad o, al menos, dificultad de controlar las reproducciones descritas y no de un «interés general » a proteger". Añadían que es por tanto muy importante estimular "las medidas tecnológicas de protección".

En la justificación de la enmienda al artículo 161 LPI dicen que el canon de copia privada, "que apoyamos plenamente", es un "remedio temporal ante una situación imposible de controlar" y que "cuando se vislumbra la posibilidad de controles efectivos, es importante que esta Ley ponga los cimientos que los hagan posibles y no que consagre un sistema nacido no para proteger intereses superiores sino como un mal menor". Ambas enmiendas las firma Eduardo Zaplana.

El 1 de Marzo de 2006, en las Cortes, la portavoz del PP dice que "solo por el hecho de que no sabemos como evitar la copia privada la regulamos y se exige que se compense al artista" para después agregar que esto puede cambiar conforme se perfeccionen las

medidas anticopia.

Muy recientemente, el 20 de Diciembre de 2007, el PP se opuso al canon en el Congreso recordando que "los tratados digitales de la Ompic que España firmó" no imponen obligatoriamente a los Estados ni el canon "ni la copia privada".

Debemos entender, mientras el PP no diga que ha cambiado radicalmente de opinión, que su postura sigue siendo la que ha venido exponiendo en los últimos años.

La postura del PSOE no es mejor. En la práctica la postura de este partido se resume básicamente en que la copia privada existe pero prácticamente vacía de contenido y como mera excusa para la pervivencia y justificación del canon. Hay que sumarle a esto las infames campañas de concienciación promovidas por el gobierno con dinero público y que han terminado por convertirse en publicidad gratuita de los productos de la industria del cine, la música y los libros.

De hecho se realizan en épocas navideñas, según aclaran ellos mismos, "coincidiendo con momentos de compra de regalos". Es decir, las campañas de "concienciación" navideñas finalizan el 5 de Enero, así los Reyes Magos se enteran bien de que tienen que traer a los niños muchos discos. Esto puede comprobarse fácilmente leyendo la memoria de 2006 de la Comisión Intersectorial para Actuar Contra las Actividades Vulneradoras de la Propiedad Intelectual, comisión adscrita al Ministerio de Cultura.

En definitiva, los que estén a favor de la copia privada y contra un canon indiscriminado no encontrarán ninguna razón para votar a ninguno de los dos partidos. El voto para uno u otro o para ninguno de los dos dependerá para muchos internautas de su afinidad con el resto del programa político, pero en lo que respecta a propiedad intelectual, hay pocos partidos, o más bien ninguno que yo conozca, que defiendan con claridad las posturas que apoyan la mayoría de internautas.

¿Recibes amenazas de alguna clase por dejar contínuamente en evidencia a los gestores de la SGAE?

No recibo ninguna amenaza. Lo que ha ocurrido es mucho más simple y aburrido: la SGAE pone como condición previa a la celebración de un debate que yo no asista.

"Copia este libro", además del tradicional formato en papel, también fue publicado en formato pdf bajo la licencia Creative Commons, y se puede descargar gratuitamente de la red. ¿Cómo conseguiste que una editorial aceptara esos términos?

Negociando con ellos. En mi caso particular no fue especialmente difícil que aceptaran porque esta editorial conocía el movimiento copyleft, así que me pude ahorrar muchas explicaciones introductorias.

Te hemos visto en tertulias, conferencias, ruedas de prensa, etc. Y hasta has colaborado en Noche Sin Tregua (Paramount Comedy). ¿Cómo acabaste allí?

Me entrevistaron en el programa con motivo de la publicación de mi libro y, un tiempo después, recibí un correo de la dirección del programa diciéndome que les había gustado mi forma de explicar las leyes y que si me apetecía volver como colaborador. Acepté encantado porque Noche Sin Tregua es uno de los pocos programas que he visto habitualmente.



MASTER SYSTEM: APRECIACIONES DE UN COLECCIONISTA

POR



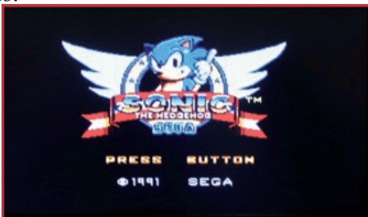
FERNANDO VERGEL

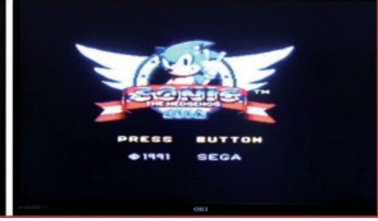
Por todos es conocido el amor de *SEGA* por el mundo del videojuego, el mismo amor que siente un quinceañero por la chica de sus sueños, tan bonito como conseguir superarse para hacer verdaderas obras de arte llenas de inspiración y sentimiento, y tan absurdo y tonto como para llegar al extremo de morir por amor. Así fue *SEGA*, una gran historia llena de despropósitos y belleza, con un montón de curiosidades que no merecen caer en el olvido.

Me gustaría hablaros de uno de los modelos mas antiguos de *SEGA*, la *SG-1000 Mark III*, más conocida por aquí como *Master System*. Casi todo el mundo ha jugado alguna vez a alguna, ya sea por emulador o jugando a la *Game Gear* (que no era más que una *Master System* portátil), y todo lo que yo pueda decir sobre sus juegos, características técnicas y periféricos, está más que dicho en Wikipedia (os recomiendo el artículo en inglés, porque es bastante completo). Yo os voy a asesorar a la hora de identificar el mejor modelo de Master System para una posible adquisición para vuestra colección, o simplemente como curiosidad y por tener un poco más de esa culturilla *freak* que tanto nos mola.

Bien, empecemos por cuál escoger. Existían varios modelos incluso dentro de la misma denominación. Dos grandes grupos, PAL y NTSC, y dentro de cada uno de estos grupos, las que usaban sintonizador de antena y las apreciadísimas RGB.

La diferencia entre las PAL y las NTSC era que las segundas tenían menos resolución vertical y casi un 20% más de velocidad de juego, así como de la melodía del mismo (realmente, todo iba más rápido). Aunque en principio puede parecer razonable perder velocidad por ganar un poco más de resolución vertical, nada más lejos de la realidad: como los juegos se realizaban en formato NTSC, la forma mas cómoda de portarlos al sistema PAL era poniendo ese "exceso" de resolución vertical del color de fondo, quedando todos los juegos "achatados" y encima más lentos que en el formato NTSC. Tal y como se ve en las imágenes.

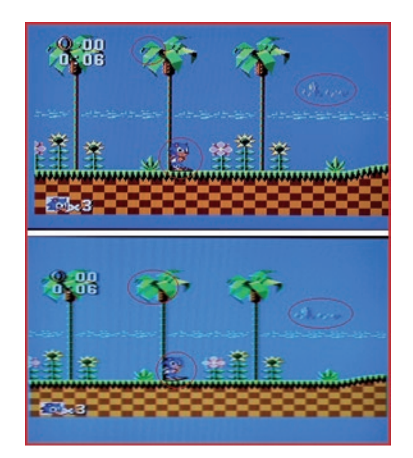








¿Llegáis a apreciar que las bandas superior e inferior en la PAL son mucho más grandes?, ¿y no notáis que los gráficos están achatados? (lo que debiera ser redondo, en una consola PAL es ovalado). Pues bien, sólo atendiendo a este detalle, haceos con una NTSC sin pensarlo (o una PAL que permita su fácil modificación a NTSC, ahora hablaremos de esto). Y quedan por ver las diferencias entre tener la consola con sintonizador de antena o con RGB. Aquí entramos en un tema que roza más el gusto de cada uno que la calidad y ahora os explico porqué, aunque mejor, juzgad por vosotros mismos con las imágenes.









Una señal RGB pura y dura no tiene prácticamente ninguna pérdida y todo se ve nítido, mientras que un sintonizador "degrada" la señal haciendo que sea borrosa y susceptible a otras influencias (como sombras de otros canales, interferencias, etc.). Tal y como lo cuento parece que está claro: todo el mundo elegiría la RGB, ¿no? Pues bien, la antena tiene un beneficio que el RGB no tiene, ¿Alguna vez habéis jugado a emuladores y viendo lo pixelado que aparece, habéis aplicado un filtro que "suavice" ese pixelado? La sintonizadora hace exactamente eso, y en algunos casos nos deja mucho más a la imaginación, que el "recio" RGB.

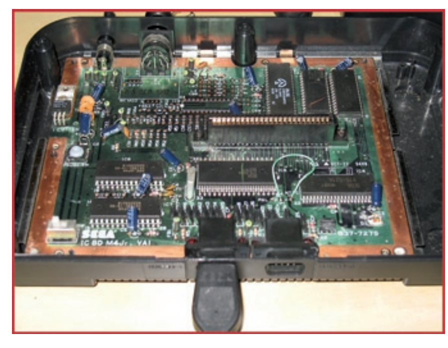
En las imágenes podéis apreciar cómo en la RGB casi se ve a *Sónico el Puercoespín* casi cuadrado; las flores son toscas y una parte de las hojas de las palmeras se ve que está compuesta de píxeles de un color y píxeles de otro. Ahora, si miráis la de antena, veréis que nos deja un poco de suavizado haciendo que Sonic nos "parezca" redondeado, las flores las vemos de otra forma (nos deja más a la imaginación) y la misma hoja de la que antes veíamos perfectamente los píxeles intentado formar otro color, aquí lo vemos perfectamente como un color nuevo que realmente no existe en la paleta del juego, aparentando que tenemos más colores de los que realmente tenemos.

La antena tiene un aire retro que creo que gustará a muchos, pero la realidad es que una vez en movimiento ya no te fijas en esos detalles, y el RGB gana sin problemas a la antena. Al RGB sólo se le puede reprochar que el conector elegido es muy duro y sólo está sujeto por las soldaduras a la placa base de la *Master System* y una avería típica de este modelo es la perdida parcial o total del vídeo o el sonido porque las soldaduras se han quebrado o la pista está levantada. Pero claro, re-soldando o haciendo un puente con cable, se soluciona.

Ahora veremos varias "tripas" de *Master System* para terminar de decidirnos por nuestro modelo adecuado.



Esta es la típica *Master system 2* de antena. Es la mas fácil de localizar y suele llevar el *Alex kidd* en memoria. Es relativamente difícil de modificar para reconvertirla en NTSC.



Esta es una "apreciada" *Master system 2* RGB. Como veis, tiene menos piezas, y si os fijáis bien, veréis mi modificación para reconvertirla de PAL a NTSC. Cualquiera puede hacerlo.



Y por último, aquí esta mi *Master System*, la clásica, este modelo es también por RGB, aunque también existe con antena. Tiene un lector de tarjetas por el cual se podían cargar juegos en un formato muy similar a las *HUCARD* de *NEC* (pero incompatibles con ellas) y también se llegó a usar con un sistema de gafas 3D para ciertos juegos. Este modelo es un poco especial porque viene con el *Hang On* en memoria y el *Snail Mace*. Pues sí, amigos, ¡esa leyenda urbana sobre un juego extra oculto en la *Master System* era cierta! Pero lamentablemente sólo es para las primeras *Master System* así que las *Master system 2* se quedaron sin el jueguecillo de marras.

Bueno, creo que como guía para elegir vuestra *Master* no está mal. Ahora sólo os queda encontrar la adecuada.




POR
SAIDA URRESTI

ANIMAL CROSSING ¡TE DAMOS LA BIENVENIDA A TU NUEVA VIDA!

¿Es un *Harvest moon*? No. ¿Es un *Sims*? ¡Por supuesto que no! ¡¡Es *Animal Crossing*!!

Aunque el nuevo juego de *Animal Crossing* para *Wii* aun está en desarrollo (estad preparados, porque la rumorología indica que saldrá este mismo año), hemos querido rendirle un homenaje a este fantástico mundo creado por *Nintendo EAD* (*Nintendo Entertainment Analysis and Development*), recordando los títulos que le preceden: *D□butsu no Mori* de *Nintendo 64*, *Animal Crossing* de *Gamecube* y *Animal Crossing: Wild World* de *Nintendo DS*.



Los juegos

El primer juego de la saga, *D□butsu no Mori* (también conocido como *Animal Forest*) no salió de Japón. Fue lanzado en Abril de 2001 para *N64*, aunque ocho meses más tarde apareció también una versión para *Gamecube* bajo el nombre de *D□butsu no Mori Plus*. Es a partir de la versión de *Gamecube* que el juego utiliza el reloj interno de la consola adaptándose a la hora local en tiempo real, cosa que es realmente importante en este juego.

A la hora de exportar hacia América este juego de *Gamecube*, ya con el nombre de *Animal Crossing*, el equipo *Treehouse* de *Nintendo of America* comenzó a traducirlo. Sin embargo, supuso muchísimo más que una simple traducción, puesto que en el juego existían frases coloquiales, eventos y fechas importantes que no coincidían con las costumbres norteamericanas. Finalmente el grupo tuvo que realizar tantas modificaciones que parecía un juego nuevo,

quedando esta versión totalmente “americanizada”. De hecho fue este juego “remasterizado” el que volvió de nuevo a Japón con el nombre de *D□butsu no Mori e-Plus*. *Animal Crossing* salió en Estados Unidos en Septiembre de 2002, *D□butsu no Mori e-Plus* reapareció en Junio de 2003 en Japón, y en Europa tuvimos que esperar hasta Septiembre de 2004.

Ya en Noviembre de 2005 (en Marzo de 2006 en Europa) sacaron a la venta en Japón la versión portátil del juego (para *Nintendo DS*): *Oideyo D□butsu no Mori*, traducido en el resto del mundo como *Animal Crossing: Wild World*.

Como ya he comentado antes, el nuevo juego de *Wii* aún no tiene fecha ni nombre, sólo se sabe que utilizará la función *WiiConnect24*, mediante la cuál se podrían mandar e-mails a los personajes o vecinos (incluso desde un PC).

En todos estos juegos la banda sonora corre a cargo de Kazumi Totaka (tienes más información en el número 3 de esta revista), que además es encarnado por *Totakeke*, un perro bohemio que ameniza la noche de los sábados con su guitarra.

El comienzo de tu historia

Tu personaje (tú, para más señas), decide independizarse, trasladándose a un pequeño pueblo. Pero eso sí, no tienes ni para pipas. El juego comienza en primera persona. Dependiendo de si estás jugando a la versión de *Gamecube* o a la de *Nintendo DS*, te encontrarás, respectivamente, en un vagón de tren (¿eso que hay al fondo del vagón es una jabalí con un cesto de nabos?) sentado frente a un gato llamado *Fran* o en un taxi cuyo conductor se hace llamar *Capitán*. En ambos casos, se te harán unas preguntas y, dependiendo de lo que contestes, tendrás uno u otro aspecto (atiende bien a las preguntas, porque definirán a tu personaje durante el resto de su vida). Cuando llegas al pueblo, te recibe *Tom Nook*, un mapache con buen ojo para los negocios. Él te alquila una casa (que deja a las famosas casas de 40 m^2 como verdaderos palacios) y te dice que podrás pagarla en cómodos plazos (básicamente, cuando quieras, ¡y sin intereses!). Después añade que, para empezar, trabajarás para él. Comienza el tutorial. *Nook* te enseña cómo plantar flores y árboles, dónde están los lugares emblemáticos (como el museo o el ayuntamiento), dónde viven tus nuevos vecinos, cómo enviar cartas o cómo decorar tu casa. En poco tiempo te ves libre de este jefe, aunque aún queda una hipoteca por pagar. Hay mucho que hacer en tu nueva vida.

Lugares y personajes importantes

Aparte de tu propia casa, hay otros lugares muy importantes en tu nuevo pueblo. Tienes la tienda de *Nook*, donde puedes comprar y vender todo tipo de cosas, como papel de cartas, semillas de flores, esquejes de árboles, herramientas, papel de pared, alfombras o muebles para decorar tu casa (o regalar a los vecinos).

También está el museo, donde el búho *Sócrates* aceptará de buen gusto tus donaciones de insectos (aunque le repugnen), peces, obras de arte y fósiles, para exponerlos públicamente.

La tienda de las *Hermanas Manitas* la regentan *Pili* y *Mili*. Allí puedes comprar ropa y complementos, ¡e incluso diseñarlos!



En el ayuntamiento puedes cambiar la melodía del pueblo (esa que suena cuando entras en la casa de alguien, o cuando hablas con tus vecinos) o enviar cartas. Es posible que en ocasiones veas al alcalde *Tórtimer* en el ayuntamiento, pero nunca podrás hablar con él ahí dentro, porque siempre está ocupado “meditando”. Es más fácil encontrarle por el pueblo en



las fiestas. En este lugar te atenderán las pelícanas *Sol* (en el turno de día) y *Estrella* (en el turno de noche). El cartero, *Carturo* no aparece nunca en el ayuntamiento, aunque es posible verle cuando te lleva las cartas al buzón.

Y aunque no es un lugar concreto, todos los pueblos de *Animal Crossing* son costeros, así que no puedes dejar de visitar la playa.

También hay algunos lugares propios de cada versión del juego. Por ejemplo el vertedero, la fuente de los deseos, la comisaría (donde se pueden "adquirir" objetos perdidos) o la estación de tren (en la que el mono *Estasio* te permite visitar pueblos de amigos metiendo una *Memory Card* con otra partida guardada en el *Slot-B*) en la versión de *Gamecube*. En la de *NDS* hay otros como el Portón (que es donde *Nocencio* y *Vigilio* controlan la entrada de visitantes y tus salidas a los pueblos cercanos, por *Wi-Fi* o de *DS-DS*, custodian los objetos perdidos y te dejan cambiar el diseño de la bandera del pueblo), la peluquería de *Marilyn* (que aparece dentro de la tienda de *Nook* cuando éste prospera en su negocio), y dentro del museo también están el bar *El Alpiste* (donde el barman *Fígaro* te recibirá fríamente, hasta que consigas entablar cierta amistad con él y llegue a ofrecerte incluso su café especial con leche de pichona) y el observatorio (donde *Estela*, la hermana pequeña de *Sócrates*, te anima a que mires las constelaciones, que puedes crear tú mismo, o puedes ver las que han hecho tus amigos en sus pueblos).



Mención aparte para la isla en *AC*. Pues sí, hay una isla tropical en la que siempre es verano. Sólo necesitas una *GBA* y un *Gamelink*, ir a hablar con el *Capitán*, que te espera en una barca, y te llevará a la isla. Incluso podrás jugar en la *GBA* con esta isla,

aun cuando desconectes el *Gamelink*. Una curiosidad es que, si pasas un buen rato en la isla, te pondrás moreno (esto también ocurre en el pueblo durante los meses de verano).

Personajes itinerantes

Sin duda uno de los más característicos es *Totakeke*. Se trata de un perro que ofrece su música a cambio de nada, ya que a él "le mueve el arte, no el comercialismo... ¡Y el arte es libre!". Puedes pedirle que te toque una canción en particular, dándole el nombre de esa canción (puedes conocer sus títulos en las casa de tus vecinos), o que te cante lo que a él le apetezca. De cualquier forma, él te regala la canción para que puedas escucharla cuando quieras en tu aparato de radio. Este personaje aparece los sábados por la noche delante de la estación (en *AC*) o en El Alpiste (en *AC:WW*).

Da Morsi también es un artista callejero. Esta morsa no ofrece música, sino papel de pared (en *AC*) o diseños (en *AC:WW*) de su propia cosecha. A cambio lo único que pide es algo "de su gusto" para comer.

Alcatifa es una dromedaria que vende alfombras (en *AC*) o te pide que le ayudes con las entregas (en *AC:WW*), dándote a cambio una alfombra o papel de pared.

Juana viene todos los domingos para venderte nabos. Ella los vende a cierto precio, y éste varía a lo largo de la semana, así que el negocio está en vender los nabos a *Nook* a un precio mayor del de la compra. El comercio de nabos ha revolucionado muchos foros de internet desde la versión de *NDS*.

Ladino es un zorro, literalmente. Es un comerciante ambulante con mercancías de dudosa procedencia. Aunque sus objetos suelen ser bastante raros, sus precios suelen ser desorbitados, y es un timador nato. *Beware the fox!*

Otros personajes comunes a las dos versiones de *Animal Crossing* son *Katrina* (la pantera vidente), *Graciela* (una jirafa con el ego por las nubes y una visión de la moda algo curiosa), *Blanca* (la gata misteriosa a la que tienes que pintarle una cara), los hermanos *Rese T. Ado* y *Forma T. Ado* (que aparecen si se te ocurre apagar la consola sin guardar, y uno de ellos tiene muy mala leche), *Gulliver* (una gaviota con mal sentido de la orientación que aparece en la playa tras un naufragio en *AC*, o es un alienígena en *AC:WW* -¿¿Quién es el loco del tirachinas?? -).



En la versión de *GC* también aparece el fantasma *Buh*, que aparece muy entrada la noche (como cualquier fantasma que se precie). Por otra parte, algunos personajes exclusivos de *AC:WW* son *Pascal* (un castor filósofo que puedes encontrar a veces a la orilla del mar), *Sisebuto* (un roedor que te visita los sábados para tima... quiero decir, para venderte



un seguro), el *Doctor Sito* (medio psicólogo, medio humorista, que te enseña expresiones gestuales), y las gatas *Catiana* y *Cati* (que aparecen a veces después de una conexión con un amigo).

Mucho por hacer

Aparte de relacionarte con todos tus vecinos y los de tus amigos, y de visitar todos los lugares emblemáticos, hay muchísimas cosas por hacer en tu nueva vida. Tienes que tener muy presente que la hora del día y el momento del año influyen en todos los acontecimientos. Por ejemplo, muy entrada la noche muchos de los vecinos estarán durmiendo, y en invierno no encontrarás demasiados insectos. Incluso influye el tiempo atmosférico, encontrando, por ejemplo, peces realmente extraños cuando hay tormenta.

En todo caso, lo primero es aprender a utilizar las herramientas. Es algo imprescindible si quieres prosperar. Todas ellas las puedes encontrar a la venta en la tienda de *Nook*. E incluso, a lo largo del juego, puedes conseguir herramientas de oro que tienen características interesantes.

Una de estas herramientas es la caña de pescar. Con ella, evidentemente, podrás pescar los peces (y otras criaturas) del río, el mar y los estanques. Ten en cuenta que cada animal vivirá en su hábitat natural, así que no esperes buscar una lubina (*¡Alucina lubina!*) en el río, ni un pez dorado (*¡qué pena que no sea de oro!*) en un estanque. La caña dorada no es fácil de conseguir, pero se supone que con ella es más fácil que piquen los peces (como si fuera difícil).

La pala es otra de las herramientas necesarias para hacer dinero. Con ella podrás plantar árboles, tanto frutales como abetos. En tu pueblo habrá cierto tipo de árbol frutal. Si recoges estas frutas para venderlas, *Nook* te dará 300 bayas por cada pieza. Sin embargo, la fruta alóctona (es decir, una que no sea la natural de tu pueblo) se cotiza mucho más (a 500 bayas). Pero si además siembras esas frutas con ayuda de la pala, podrás conseguir árboles frutales nuevos, y con esa producción conseguirás mucho más dinero. La pala de oro sin duda es la más fácil de conseguir, y sirve para... bueno, ¿sabes eso de que el dinero no crece en los árboles?

El hacha tampoco es muy necesaria, excepto para tu búsqueda del pueblo perfecto (plantar árboles aquí, quitar de allá). Lo malo de esta herramienta es que se rompe con facilidad, cosa que no ocurre con su homóloga de oro.

Con la red podrás cazar todo tipo de insectos y arácnidos. La red dorada tiene unas dimensiones considerablemente mayores que la normal, con lo que los bichos se cazan más fácilmente.

Una de las novedades de *AC:WW* es el tirachinas, con el que podrás derribar cualquier objeto que vuele por el cielo. La ventaja que da el tirachinas dorado es que en lugar de un tiro, realiza tres a la vez.

Otra novedad de *AC:WW* es la regadera. No es del todo imprescindible, pero ayuda mucho a mantener lleno de flores el pueblo.

Pero hay muchísimas más cosas que hacer en *Animal Crossing*. Puedes cartearte con tus vecinos y amigos. Puedes ampliar tu casa y decorarla a tu gusto. Puedes colocar mensajes en el tablón de anuncios. Puedes personalizar la melodía de tu pueblo. Puedes conocer las historias de los personajes que vas conociendo (en *AC:WW* hay algunos personajes que en ocasiones necesitan un oído que les escuche). Puedes coleccionar series de objetos, u objetos raros (a veces *Nintendo* ofrece objetos de coleccionista a los que se conectan por *Wi-Fi* en ciertas fechas). Hay muchísimas alusiones a otras grandes sagas entre estos objetos raros: *Mario*, *Zelda*, *Pokemon*, *Metroid*, *Pikmin*, etc. En *AC* incluso puedes conseguir juegos de *NES*, como *Soccer*, *Clu Clu Land*, *Donkey Kong*, ..., que además de jugarlos en tu televisión, también puedes descargarlos a tu *GBA*.

Eventos y fechas importantes

Como en *Animal Crossing* los días pasan de la misma forma que en la vida real, también hay este

paralelismo con ciertas fechas señaladas del calendario. En algunas de estas ocasiones se pueden conseguir objetos raros, o conocer personajes exclusivos de esas fechas. Según la versión del juego (recuerda que *AC* está muy americanizado, mientras que *AC:WW* es bastante más neutral), podrás celebrar eventos como Año Nuevo, San Valentín, el Día de los Inocentes, el Día de la Madre, el 4 de Julio, *Halloween* (en el que deberás encontrar a *Soponcio* para conseguir objetos de la serie Calabaza), el Día de Acción de Gracias (en el que tienes que salvar al pavo, *Guindo*, de ser comido), Navidad (en el que puedes encontrar a *Renato* y pedirle objetos de la serie Festiva), Fin de Año, etc. También hay otros eventos exclusivos de este fantástico mundo, como el Día del Deporte, el Día de la Naturaleza, las competiciones de pesca y de caza de bichos, el Día de la Luna de Otoño (que por cierto, en inglés llaman a este día *Harvest Moon*), o la Semana de la Bellota (en la que tendrás que ofrecer bellotas al misterioso *Córnimer* para obtener objetos de la serie Hongo). Además, en Agosto (sólo en *AC*), *Vigilio* organiza cursos de aeróbic (¿¿quién se levanta a las 6 de la mañana en verano para hacer aeróbic??), en invierno puedes hacer muñecos de nieve, y en diferentes fechas a lo largo de todo año se celebran mercadillos y otros eventos sociales (en *AC:WW*).



Animal Crossing vs. Animal Crossing: Wild World

Aunque ya hemos comprobado que hay numerosas diferencias entre ambos juegos, sin duda la mayor comparativa que se puede establecer es respecto a su jugabilidad. Ninguno de los dos es que sea mejor o peor que el otro, puesto que ambos tienen cierto encanto que hace que te sientas enganchado a jugar al menos un rato cada día. Pero si nos ponemos a comparar, una de las grandes ventajas de *AC:WW* es precisamente su pantalla táctil. Por ejemplo, en *AC* la experiencia más tediosa es escribir cartas. Esto es muchísimo más fácil con el teclado táctil de *AC:WW*.

Algo muy interesante en la versión de *NDS* es que cuando visitas a tus amigos





en sus pueblos, existe la posibilidad de que sus vecinos se trasladen a tu pueblo. Lo más curioso es que tienen memoria. Recuerdan perfectamente que les conociste en ese pueblo, recuerdan al amigo que "os presentó", recuerdan la melodía de su antiguo pueblo... También tus propios vecinos pueden preguntarte sobre amigos que les hayan visitado en tu pueblo.

Ya para finalizar, no puedo dejar sin nombrar la película de *Animal Crossing*. Así como lo lees, hay una película de animación de *Animal Crossing*, aunque ni ha salido de Japón, ni parece que lo vaya a hacer. Sobre el argumento... no difiere mucho del que ya he contado. Y bueno, aunque podría estar media vida contándote más secretos de estos juegos, creo que es mejor que los pruebes y los encuentres por ti mismo.



DOUBLE WIIMOTE BATTERY PACK [DRAGON]







Una de las cosas más chocantes del inverosímil mando de *Wii* es el uso de pilas comunes para su alimentación. Realmente, no comprendo cómo *Nintendo* no puso algún tipo de batería en el mando, aunque también es cierto que en caso de incluirla se hubiera encarecido el coste final del producto (ya sea la consola completa con un mando, o los mandos por separado), y teniendo en cuenta que el precio es uno de los ganchos de *Wii* (que 250€ parezca un precio bajo no habla demasiado bien de la competencia), parece normal que no incluyeran la alimentación del mando en el paquete.

Para paliar estas inconveniencias, han surgido una tonelada de dispositivos de entre los cuales vamos a analizar éste: el pack de dos baterías de *Dragon*.

¿Cómo se alimenta una batería dentro del *Wiimote*? Dado que aparentemente no hay forma de suministrar la carga a través del puerto de conexión del mando, las compañías como *Dragon* han optado por usar pilas recargables (dos de tipo AAA de 750 mAh en el caso que nos ocupa) conectadas al exterior mediante una tapa para el mando modificada. Los contactos al exterior hacen posible el suministro de carga mediante una base plástica que acoge dos mandos.

Hablando de dicha base, podemos decir que es de color blanco, con una textura muy agradable y que no desentona con la sofisticada estética de la consola. Presenta un par de agarres que se usan para fijar el mando, y que permiten que éste se pueda abatir contra los contactos (y de esa manera recargarlo) o dejarlo en posición vertical sin que se produzca carga alguna. Esto último es algo muy práctico debido a que las baterías que permanecen siempre conectadas a un circuito de carga terminan deteriorándose, y resulta interesante poder dejar los mandos en posición vertical para su almacenamiento.

En cuanto a la duración, es bastante complicado de estimar, puesto que depende mucho de los juegos que se usen. En las pruebas que hemos hecho ha sido posible jugar durante toda una tarde sin que se terminaran de descargar las baterías. También, debido a que dichas baterías son estándar, es posible sustituirlas por unas con mayor capacidad (1.000 mAh, por ejemplo) para obtener un extra de rendimiento.

Otra de las ventajas de este aparatito es que la alimentación proviene de un puerto USB de la consola. Al menos las terceras empresas usan para algo los puertos USB de *Wii* (EA los usó para conectar un micrófono), porque todavía no sabemos cuánto más vamos a tener que esperar para ver algo de la mano de *Nintendo*.

Sólo le hemos encontrado dos pequeños fallos, si es que se les puede llamar así. La base de carga tiene dos *LEDs* para indicar que las baterías se están cargando. Pues bien, el primer fallo, es que esas brillantes lucecitas son rojas, rompiendo un poco la estética de *Wii* (blanco con luces azules). Y por otra parte, una vez cargadas las baterías, el *LED* se pone a parpadear incesantemente, lo que en ocasiones puede ser bastante incómodo. Sin embargo, solucionar esto último es tan sencillo como dejar vertical el mando para evitar el contacto de la batería con la base.

¿Veredicto final? Si no tienes nada con qué sustituir tus contaminantes e incómodas pilas alcalinas, hazlo con esto. Caben dos mandos, vienen baterías, se conecta por USB y es bonito. Y además es extremadamente económico.

¿Una nota del cero al diez? Lo siento, te has equivocado de revista.





POR









GUITAR HERO 3 [WII]



Hace ya unos meses que salió a la venta la última entrega de la conocida saga rockera *Guitar Hero*. Este tercer capítulo no presenta demasiados cambios si lo comparamos con sus predecesores que tan buenos resultados han dado, por lo que la máxima *"si algo funciona no hay porqué cambiarlo"* tan utilizada en la industria del entretenimiento audiovisual en este caso también se sigue al pie de la letra. Vamos, que es el mismo juego con la cara lavada. Dicho esto, en ningún caso hay que desmerecerlo ya que resulta entretenido e incluso adictivo, que tal y como está la cosa no es moco de pavo.

Esta tercera entrega, que lleva por nombre *Guitar Hero III: Legends of Rock* (*Activision*, 2007), es la primera que aparece en una consola de *Nintendo*. El juego se vende conjuntamente con una réplica de una guitarra *Les Paul* de *Gibson* a la que se le conecta el *Wiimote*, con lo que tenemos la sensación de estar tocando una guitarra de verdad tanto por la fidelidad de la guitarra a la original como por las ventajas de tener el mando integrado en la misma (sonido y vibración propios).



El juego empieza con una multitud de videos de las diferentes compañías que han participado en el proyecto tanto en el diseño del *software* como de la guitarra (en total 6) y una vez en el menú principal vemos que hay bastante variedad de modos de juego (carrera, carrera cooperativa, partida rápida, uno contra uno, modo *online*,…) para el mismo concepto de juego. Dicho concepto básico es muy sencillo: utilizando la guitarra (también se puede jugar con el *Wiimote* pero es mucho menos intuitivo y sobre todo menos divertido) hay que tocar el mayor número de notas correctamente siguiendo el orden y el tempo de las bolas de colores que bajan por la pantalla.

Parece sencillo, ¿no? Tal vez sí, tal vez no. Para los más novatos, se incluye un modo de tutorial que puede llegar a ser interesante si se le quiere sacar todo el jugo al juego. Además, para los más precavidos hay una opción de practicar las canciones antes de empezar a jugar "en serio". Puedes tocar cualquier parte de la canción, desde el estribillo a uno de esos solos de guitarra que harían levantarse de su tumba al mismísimo Jimi Hendrix.

Si decidimos comenzar nuestra experiencia musical en la modalidad de carrera tendremos que ir superando varias canciones divididas en bloques, cada uno con su escenario propio (hasta hay una cierta "historia" dentro del juego). Mientras no falles todo va bien y puedes ir subiendo tu puntuación gracias al multiplicador que va subiendo hasta un "x4", pero cuando fallas una nota esto se acaba y sale un sonido chirriante del altavoz del *Wiimote*. Si fallas demasiadas seguidas puedes entrar en la zona roja del indicador de nivel e incluso puedes no acabar la canción (algo muy raro de ver en nivel fácil, por cierto).

Otro detalle del juego consiste en acertar una serie consecutiva de notas marcadas con estrellas con las que se va llenando un contador. Cuando has encendido suficientes bombillas, puedes activar la Energía Estrella subiendo la guitarra hasta ponerla casi vertical aprovechando las características del *Wiimote* (gracias a que todo esto lo hace el mando de *Wii*, la versión para esta consola cuesta 10€ más barata). Con la Energía Estrella activada se distorsiona ligeramente el sonido y toda la guitarra vibra. Es una sensación extraña pero que (en ocasiones) se agradece. En este momento es cuando el multiplicador se dobla para conseguir aun más puntos.

Cuando conseguimos terminar un número de canciones por bloque (3 en los modos Fácil y Medio, ó 4 en el Difícil y el Experto) nos pedirán tocar un *bis* o incluso una Guerra de Guitarras contra guitarristas famosos como Tom Morello (de *Rage Against the Machine*) o Slash (de *Guns N' Roses*). Al acabar el último bloque nos veremos retados por un último invitado que, aunque la primera vez parezca un poco ridículo, se le acaba cogiendo cierto cariño. Y hablando de cariño, la canción que se desbloquea







cuando nos pasamos el juego es de la mejores. Hablamos de la epiléptica *Through the fire and flames*, del grupo *Dragonforce*.

Por desgracia, las canciones (hasta que *Activision* decida cambiarnos los discos de una vez) están codificadas en *mono* y teniendo el sonido en modo *envolvente* suenan como si dos gatos se pelearan dentro de una lata. El apartado gráfico del juego (os aseguro que lo único que vais a ver van a ser las bolitas de colores, incluso con los ojos cerrados) acompaña lo suficiente y es correcto pero sin grandes alardes. Sin embargo, si te quedas de espectador mientras algún amigo tuyo juega, y te fijas en los personajes o en los acontecimientos del concierto, puedes ver cosas curiosas, como por ejemplo que el cantante abre más o menos la boca según lo que está cantando, o que los guitarristas mueven las manos más rápido o más despacio según las notas que estén tocando, aunque esto ya ocurría en anteriores juegos de la saga.

En cuanto a los otros modos de juego, se agradece la existencia de un modo cooperativo. En este modo puedes jugar con un amigo, uno tocando la guitarra y otro el bajo (o uno como solista y otro de ritmo acompañante, si se trata de una canción en la que no hay bajo) y tiene el aliciente de que hay canciones nuevas que no encontrarás en el modo individual. Tendrás que jugar por parejas para tener todas las canciones disponibles en "partida rápida". Yo soy

de la opinión de que siempre es mejor y más divertido un modo cooperativo que uno competitivo, ya que es mejor hacer fuerza hacia el mismo objetivo que andar zurrándose uno contra otro (con tanta competitividad luego hay gente que ve armas de destrucción masiva por todas partes).

El juego se completa con otras opciones, como una tienda donde poder comprar guitarras, canciones, videos e incluso personajes (la caricaturización de la peor versión de Elvis no tiene precio) y otros modos de juego como el *online*. En lo que respecta a las canciones que incluye el juego, se podría decir que para gustos están hechos los colores pero la palabra que mejor define la colección musical del juego es heterogeneidad. Hay una mezcla de canciones clásicas (*Kiss*, *Rage Against the Machine*, *The Rolling Stones*, *The Smashing Pumpkins*, *Black Sabbath*, *Pearl Jam* o *Metallica*) con otras más actuales (*Muse*, *Bloc Party*, *The Strokes*, *Queen of the Stone Age* o *The Killers*) pero en general su calidad es buena. En cuanto a las canciones que se pueden comprar en la tienda (incluida una de *Héroes del Silencio*) son en su mayoría mucho más recientes y, por qué no decirlo, más (ir)regulares. También se agradece que entre las canciones "extras" se incluyan algunas en francés (*Superbus* y *Naast*), alemán (*Revolverheld* y *Die Toten Hosen*) o castellano (los nombrados Héroes del Silencio), ya que aumenta aun más la heterogeneidad.

Resumiendo, se podría decir que estamos frente a un buen juego que no aporta demasiadas novedades al género, pero que gracias a la guitarra nos permite meternos de lleno en nuestra "misión". Además los diferentes modos de juego y niveles de dificultad nos aseguran un buen número de horas de entretenimiento y diversión. Eso sí, si se llega a extremos insanos (daos una vuelta por *YouTube*) creemos que es mucho más recomendable y gratificante aprender a tocar la guitarra de verdad.





POR



IKER UNAI
LARBURU





REFLEXIONES DE UN WANNABE

Hablar de *hackers* es como hablar sobre la posición y velocidad de las partículas subatómicas. No, no es ninguna broma; tal y como al hablar de los electrones, cuanto más sabemos de su posición menos sabemos de su velocidad, al hablar de *hackers* nos guiamos por el mismo principio de incertidumbre: cuanto más investiguemos a estos sujetos, más dudas tendremos sobre ellos. Cuanto más *hacker* es uno, menos habla de ello con *no-hackers* (supongo que estos serán como los *muggles* de *Harry Potter*). Y dado que es necesario conocer el mundo *hacker* para ser uno de ellos, los neófitos estudiamos este mundo como cualquier otro efecto físico invisible sólo medible por los efectos que provocan: desarrollando teorías que son válidas mientras no se demuestre lo contrario.

Con la intención de facilitar las cosas a aquellos que, como yo, quieren introducirse en este cibermundo, paso a relataros las teorías que he conseguido establecer tras años de estudio de estos esquivos individuos. Todos tendríamos que dar las gracias por la existencia de películas como *Hackers* u *Operation Swofish*, auténticos ejemplos de investigación social que tanta luz han arrojado sobre este mundo.

El nick

Como todos los grandes artistas, los *hackers* quieren ser recordados por sus grandes obras. Sin embargo, como para el común de los mortales son poco más que delincuentes cibernéticos (puede que incluso *pendejos*, si además de *hackear* se bajan música), tiene que ocultar su nombre con un mote o *nick*.

El *nick* de un hacker es su mayor tesoro, tanto que puede dedicar semanas a crear uno. Estos *nick* suelen ser dos palabras escritas en el idioma de Shakespeare que tienen que cumplir alguna de estas tres características (cuantas más mejor): ser pegadizo, estar relacionado con el mundo de la informática y dejar claro lo malos que pueden llegar a ser si se cabrean.

Algunos ejemplos famosos de *nicks* podemos verlos de la película *Hackers*: *Zero Cool*, *Crash Override*, *Cereal Killer* o *Acid Burn*. De estos, solo el segundo cumple con las tres máximas, pero aun así son ejemplos verdaderamente representativos.

La "frase"

Cuando dos *hackers* se enfrentan en singular duelo cibernético es imprescindible que cada uno tenga una frase lapidaria con la que cerrar la justa. Esta frase es tan importante como el propio *nick*. Si uno de ellos es capaz de dejar colgado el sistema del otro mostrando su frase lapidaria en pantalla, el otro no tendrá más remedio que admitir su derrota y realizar un *seppuku* o suicidio ritual.

Para esta frase hay que reconocer que el idioma anglosajón vuelve a mostrarse superior al castellano tradicional. No hay más que ilustrar esta afirmación con el siguiente ejemplo. Mientras que la frase *"si te metes con el mejor, muere como los demás"* suena de lo más triste en castellano, *"Mess with the best, die like the rest"* tiene una cadencia prácticamente poética que nos hace agachar la cabeza y reconocer el genio creativo del autor.

La forma de hablar

Este es otra de las grandes barreras a salvar si se quiere llegar a ser *hacker*. Tienen un idioma a parte, compuesto de siglas y tecnicismos que es difícil de seguir, incluso, para iniciados en la materia. Muchos informáticos comunes quedarían ciertamente desorientados escuchándoles.

Pero... ¿quién no siente respeto por gente que puede decir frases como esta sin pararse a pensar en lo que esta diciendo?: *"El otro día entré en el servidor HTTP de la CIA mandando un datagrama codificado mediante un TripleDES al puerto de control FTP para provocar un stack overflow en el proxy principal de manera que al ejecutar el script de reset insertara el código de un gusano que había metido en el vector de interrupciones"*. Hala.

La metodología

Después de todo esto, la pregunta que pasa por nuestras cabezas es: ¿cómo consiguen entrar en cualquier máquina? La respuesta a esta pregunta es más simple de lo que parece. Si algo sabemos gracias a las películas es que todo sistema informático está protegido por una contraseña con algún significado de relevancia para el usuario del sistema; así, si se le olvida, podrá recordarla utilizando un razonamiento lógico.

Así que si queremos acceder al sistema de cualquier persona, lo importante es estudiar a esta persona y realizar un proceso lógico-deductivo a partir de los datos. Por ejemplo, si sabemos que la fecha de cumpleaños de una persona es 14 de febrero la contraseña más probable será *latrodectus*. A esta conclusión se llega siguiendo este sencillo razonamiento: como el usuario ha nacido el 14 de Febrero, es decir, el 14-02 utilizara este dato para recordar. Si unimos los números obtenemos el año 1402, que casualmente es el año en el que se dio la Batalla de Angora. De Angora son famosos los gatos. Los gatos de Angora tiene uñas, luego arañan, y la araña más conocida es la viuda negra. Y, para cerrar el razonamiento, el nombre científico de la viuda negra



es *Latrodectus tredecimguttatus*.

Hay otra característica importante a tener en cuenta para ser un buen *hacker*. No basta con poder realizar este tipo de razonamientos: hay que hacerlos sin despeinarse, y en las circunstancias más adversas. El protagonista de *Operation Sworfish* es capaz de colarse en la página del FBI en un minuto después de estar intoxicado con un grado de alcohol digno de Boris Yeltsin, mientras 3 matones le han amenazado de muerte y con una meretriz practicándole una felación mientras está en ello. Yo por mi parte mantengo que, con lo bueno que era este tío, se permitió el lujo de dejar trabajar a la chica durante 50 segundos y, después de quedar satisfecho, hackeó el sitio Web en los 10 segundos restantes. Pero es una teoría.

El equipo

Por último queda decir que los *hackers* construyen sus propios equipos. Tienen tal amor por sus equipos que los personalizan de tal forma que se puede encontrar poco tanto del *hardware* como del *software* original. La potencia de estos equipos es tal que sacarían los colores a los ordenadores del *top500* (www.top500.org).

Estas máquinas suelen tener pantallas enormes o incluso múltiples pantallas, pues todo el mundo sabe que la potencia de un ordenador es proporcional a la superficie de la pantalla del mismo. De hecho, la pantalla es la pieza clave del mismo. Puedes reducir a cenizas los demás componentes de la máquina, que mientras no destruyas el monitor, seguirá funcionando.

Bueno, creo que esto es todo. Os dejo, que voy a intentar desarrollar ese sistema 3D virtual que permite navegar por el ciberespacio como si de un simulador de vuelo se tratara, intentando esquivar las defensas de los distintos sitios Web. Una vez lo haya desarrollado lo utilizaré para convertirme en el terror de la red, el temido *raw restart*, y os llenaré de conejos vuestras malditas peceras.





PHOENIX WRIGHT: ACE ATTORNEY TRIALS AND TRIBULATIONS

